Journal of Genetic Genealogy

Journal: www.jogg.info
Originally Published: Volume 10, Number 1 (Fall 2022)
Reference Number: 101.004

CALCULATING AUTOSOMAL DNA MATCH COVERAGE: A GENERALIZED ADDITIVE RECURSIVE METHOD

Author(s): Wesley Johnston

Journal of Genetic Genealogy

Calculating Autosomal DNA Match Coverage: A generalized additive recursive method

By Wesley Johnston, June-July 2022

Overview

An autosomal DNA-tested descendant of a person matches some people who an ancestor would have matched if the ancestor had tested. We can calculate a precise theoretical[i] percent of the ancestor's matches who would match the descendant. For example, a child matches about 50% of the people her father would have matched if her father had tested.[ii]

If multiple descendants of an ancestor had autosomal DNA-tested, they together match a higher percentage of the ancestor's potential matches. We can calculate this percentage. For example, five children of she same parent would match about 96.875% of the people who would match that parent if the parent had tested.

Paul Woodbury coined the term "coverage" for this calculated percent and developed formulae for doing the calculation in parent-child scenarios.[iii] His formulae do not easily extend to a generalized additive way of recursively calculating coverage that allows for automation of coverage calculation for any given combination of descendants.

This paper presents a generalized additive recursive method that allows for automated calculation of autosomal DNA match coverage of an ancestor. This paper then provides a visionary glimpse of a related tool to automatically calculate the coverage of a specific relative from the DNA of other family members towards identification of the remains of a family member lost in a prior conflict.

I intend this presentation of the method to allow anyone to implement the pseudo code in actual code. Developers can implement this conceptual version in a great many different ways. The first developer to implement it is David Stumpf who has implemented it as a feature in his Graphs for Genealogists software (https://www.wai.md/gfg).

Basis of Calculation

The calculation rests on the generalized extension of basic coverage. One child covers 50% of a parent's autosomal DNA matches. Two children (other than identical twins) combine to cover 75% of the parent's matches. (See Figure 1.)

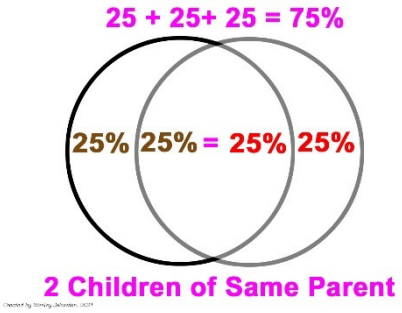


Figure 1-Two Children

Each child inherited 50% of the parent's autosomal DNA. Each child shares about half their DNA inherited from that parent (thus 25% of the parent's DNA) with the other child. But each child has a unique half (25%) that the other child did not inherit.

For example, Amy and Ben both inherited about the same 25% of their mother Mary's DNA, but Ben does not share about 25% of Amy's DNA that she inherited from Mary in locations where Ben inherited his DNA from their father Frank. And Amy also does not share 25% of Ben's DNA that he inherited from Mary because in those locations Amy inherited her DNA from Frank.

Thus, the two children Amy and Ben would match about 75% of the people who Mary would have matched if Mary had tested. The 75% comes from the shared 25% plus the two unique 25% regions of DNA each child inherited: 25% shared + Amy's unique 25% + Ben's unique 25% = 75%.

This generalizes as more children's tests add to the coverage of a parent's autosomal DNA matches. (See Figure 2.)

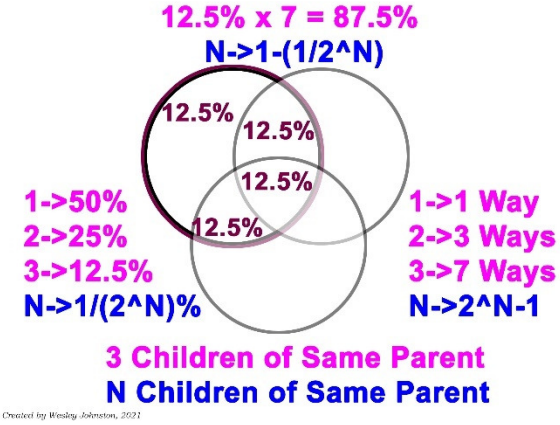


Figure 2-3 or more Children

The sizes of the shared and unshared regions of DNA reduce by half with each new child. And the number of shared regions increases as the number of children increases.

In general, N children cover the matches of about 1 – (1/2^N) of the parent (expressed as a percent, which is 100 times the number from the formula). So, three children cover 87.5% of the parent's matches.

And we have (2^N)-1 separate pieces of the parent's DNA that exist for N children, either as unique to each child or as some combination of 2 or more of the children. So, three children combine in 7 (=8-1) different pieces.

And each of those pieces contains 1/(2^N) of the parent's DNA. So, each piece for 3 children holds 12.5% (1/8 expressed as a percentage) of the parent's DNA.

The question then is how to extend this to a generalized recursive method that allows for automation of any combination of descendants – not just children but grandchildren or even more distant descendants.

Two-generation Calculation

Instead of two children testing, what is the coverage of a target person if the two tests are a child and a grandchild? The grandchild must be the child of a second child since the child of the first child adds nothing to the coverage of the parent since that child would have inherited all their target person's DNA from their parent. (See Figure 3.)

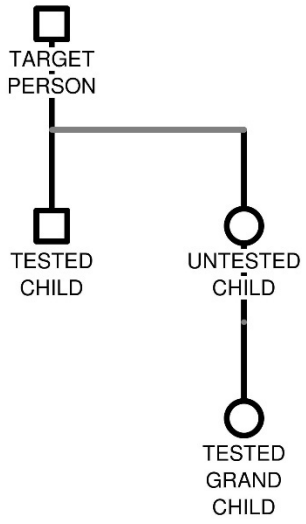


Figure 3-Uncle and Niece Tested

Another way to view the same situation is that the testers are an aunt/uncle and a niece/nephew. The niece/nephew only inherited 25% (half of 50%) of the grandparent's DNA. So, we have a child who inherited 50% of the target person's DNA and a grandchild who inherited 25% of the target person's DNA.

So, we go back to figure 1. The tested child provides their own unique 25% plus 25% that the child would have shared with the grandchild's parent. But the grandchild provides only 12.5% in each case and not 25%. In the case of the shared DNA, the tested child has provided a full 25%, so that there is no loss from the sharing with the grandchild.

The result is that we have the child's unique 25%, the grandchild's unique 12.5% and the 25% that the child and the parent of the grandchild shared. So, the child and grandchild would match 62.5% of the people who the target person would match if the target person had tested: 62.5% = 25% + 12.5% + 25%.

This is the first extension of the method.

The Goal: A Recursive Additive Algorithm

The extension above differs from the formulae that Paul Woodbury developed. His formula had negative terms that subtracted some amount in the calculation. That formulation made it very difficult to extend to all situations so that a computer algorithm could receive any configuration of descendants and generate the precise percentage of coverage of the target ancestor. The method in the previous section adds components together to calculate the coverage. It has no negative terms.

The ultimate tool would accept as input the specification of the relationships of all tested descendants and then provide the percentage of coverage of the target ancestor by the tests of those descendants. The DNA Painter website's WATO (What Are The Odds?) tool graphical user interface (GUI) is very close to what is needed, although it does not yet permit specification of descendants of the target person who married each other.

For example, here is the same configuration used in the previous section but specified with the WATO GUI.

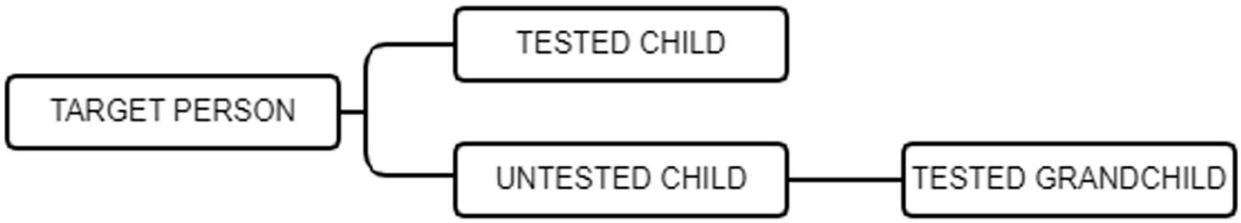


Figure 4-Uncle and Niece Tested - WATO GUI Specification as Input to Coverage Calculation Algorithm

This would be the input. The output would then tell you that the coverage of the target person. This would not only be a wonderful tool for knowing how much coverage your existing kits provide for an ancestor. It would also allow you to evaluate which of several candidate testers would contribute most to increasing the coverage.

But how does the computer do the calculation? What exactly do we tell the computer to do?

What I propose is a recursive method where each generation is another iteration that uses the same algorithm. Using an additive method allows for recursion.

Conceptual View of the Recursive Additive Algorithm (Pseudo Code)

Essentially, the algorithm generates an N dimensional Venn diagram in tabular form, with a row of he table for every distinct piece of the Venn diagram. The algorithm then calculates the coverage of the target person contributed by each piece and then sums the value of the pieces to arrive at the coverage of the target person by that configuration of DNA-tested descendants.

The algorithm would implement the following pseudo code, using the tables and variables indicated. The reference person would change with each call of the module. It would begin with the overall target person as the reference person, but the reference person in lower level calls would be a child or grandchild or other descendant of the target person as the recursive calls of the module handle each generation.

The module implementing the algorithm uses two tables to keep track of (1) the pieces of the Venn diagram and (2) the reconstructed DNA percentage of each child of the parent and then combine these into a coverage value for each piece of the Venn diagram which add up to a sum that is the autosomal DNA match coverage of the reference person by the tests of his/her descendants.

Appendix 1 steps through the calls of the module for an example application. Appendix 2 shows how the algorithm outputs the same result from different orders in which the children are presented to the algorithm.

The Five Variables

The module assigns the value of the first variable N by counting the number of children of the reference person. It then calculates variables P and M from the N. Variables W and R are indexed sub-arrays of Table 1, calculated by the module.

N = count of the number of children of the reference person (input is the tree specification)

P = number of pieces of the Venn diagram for N children = (2^N) – 1 (lower case “p” then becomes the index for Table 1, starting at 1, and lower case “n” becomes the index for the child columns)

M = maximum theoretical coverage of each piece = 1/(2^N) as either a percentage or number

W(p) = weight of the actual reconstructed coverage of each piece (see the pseudo code for details)

R(p) = result for each piece = M * W(p)

N, P and M are all fixed for the duration of a call of the module. W is the key variable that applies the calculated weight to each piece.

N and P determine the dimensions of the tables. M, W and R are specific to the calculation of coverage.

The Two Tables

It is important to note that these two tables exist for not only the top-level call for the target person. Every recursive call for every parent-child scenario has its own Table 1 and Table 2. The size of the tables at any particular call to the algorithm is determined by the number of children in that particular parent-child scenario. There will be many Table 1s and many Table 2s generated in fulfilling all the recursive calls in order to calculate the final Table 1 and Table 2 at the top-level call of the module.

Table 1 – People and Pieces (N+3 columns x P rows)

The People and Pieces table has a column for each child and a row for each piece. Table 1 is simply the tabular representation of a Venn diagram for N children, extended to include the calculation of the contribution of each piece.

The code initializes the table and never changes the columns for the children (1 to N) and M. The module changes only the cells for W and R. Each child-column cell has a binary 0 or 1 to indicate the inclusion or absence of that child in that piece. So, the values of the child-cells in a row are simply the binary equivalent of the index (starting at 1) for that row.

Here is an example of the completed (just prior to calculation of the coverage for the reference person and exiting the module) Table 1 for the first call level (the level at which the target person is the current reference person for that call of the module) of the two-generation scenario shown in Figure 4, with child 1 as the tested child and child 2 as the untested child. Since there are two children, N=2. Then P = (2^N) – 1 = 3. And M = 1/(2^N) = ¼ = 25% = 0.25.

Table 1 - People and Pieces
With Completed W and R Values in Call Level 1 with input from Figure 4

Piece\Child	1	2	M	W	R
1	0	1	25%	50%	12.5%
2	1	0	25%	100%	25%
3	1	1	25%	100%	25%

The table certainly can be modified or even broken into separate tables, as someone prefers for their implementation. For example, since the value of M is the same for every row, the table really does ` need a column for M. I include M for clarity, but since the variable M already has this value, the column for M can be omitted. In fact, a hyper-simplified version could simply use the two arrays W(p) and R(p) since the child binary combinations can be computed from the index value (e.g. p=3 is binary 11.)

Table 2 – Tested/Reconstructed DNA of Children (1 Row x N columns)

The table of Tested/Reconstructed DNA of Children has a column for each child. The cell value is the calculated percentage of coverage of that child's own DNA exists in the DNA tests of him/herself or his/her descendants.

Table 2 – Tested/Reconstructed DNA of Children
Final Result for Call Level 1 with input from Figure 4

1	2
100%	50%

Pseudo Code: Recursive Module

1. Count the number of children of the reference person (in the first call, this is the target person). Set this as N. If N=0, return the coverage percentage as 100% and exit the module.
2. Calculate the number of pieces P of DNA (as in Figure 3) and the maximum coverage percentage M that each piece holds of the reference person's DNA.
3. Generate the module-internal tables, setting values to be calculated initially at zero.
4. For each child, determine the coverage of the child by calling this same module.
5. For each piece/row of Table 1, calculate W and R for that piece/row. (details in next section)

6. Add the percentages for the pieces together to calculate the coverage percentage of the reference person and return that number and exit the module.

Pseudo Code: Step 5 calculations

W(p) = Max of all products of [cell (p, n) * M] for row p

R(p) = M * W(p) for row p

Each weight W is calculated for its row by using the combination of children specified in the binary columns for that row and multiplying that 0 or 1 value by the Table 2 value of the tested/reconstructed coverage of the DNA of that child. The maximum value of each of these products then becomes the value of W(p) for that row p. See the example in the appendix for details of how this is applied.

How to Understand the Tables and the Implicit Venn Diagram

Remember that the target person and every one of their descendants on a descendant line that has a DNA tested descendant will generate a call of the module. There will be a Table 1 and a Table 2 for every call of the module.

In cases where at least one child of the target person has tested, a single number can represent coverage, and the algorithm in this paper uses that number.

In cases where no child of the target person has tested, only a range of coverage from a lower bound to an upper bound can fully represent the reality of the coverage. The algorithm in this paper calculates the lower bound.

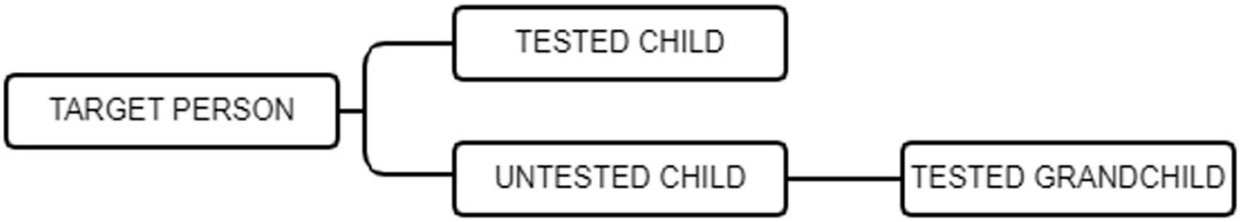
Other methods (such as Paul Woodbury's formula) calculate an average value. But using that average from one generation to calculate the coverage in the prior generation (recursive use of the formula) skews the final result away from the actual average of the final upper and lower bounds. Only the lower and upper bounds work in a recursive application of a method to accurately calculate the coverage of the prior generation when it is a range and not a single value.

See Appendix 4 for a full discussion of the handling of combinations of DNA results when those results are in fact ranges and not single values.

Table 1 is a tabular form of the Venn diagram. Each row of the table corresponds to a piece of the Venn Diagram. So, it is crucial to the understanding of what the algorithm is doing to understand what the tables are showing and how they reach their final total.

Figure 4 Scenario

To recapitulate, the input in Figure 4 is a person who has one child who has tested and one grandchild who has tested.



And the final results for Tables 1 and 2 are as follows.

Table 1 - People and Pieces
With Completed W and R Values in Call Level 1 with input from Figure 4

Piece\Child	1	2	M	W	R
1	0	1	25%	50%	12.5%
2	1	0	25%	100%	25%
3	1	1	25%	100%	25%

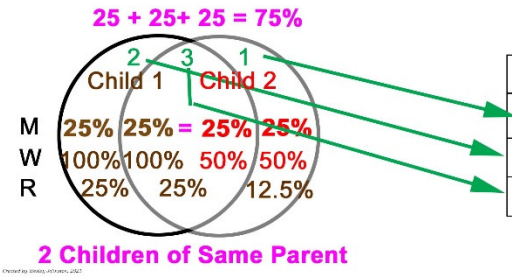
Table 2 – Tested/Reconstructed DNA of Children
Final Result for Call Level 1 with input from Figure 4

1	2
100%	50%

So, precisely what are the tables telling us?

Table 2 tells us that Child 1, the child who tested, has 100% of their DNA results for coverage of the parent. And Child 2, the child whose child tested, has 50% of their results available. Table 2 tells us how much of each child's DNA they can contribute to the parent's coverage.

Table 1 is the Venn diagram for the 2-child scenario, with the calculations completed for each piece of the Venn diagram. Let's look at what this chart spells out.



Child 1 Child 2

Piece\Child	1	2	M	W	R
1	0	1	25%	50%	12.5%
2	1	0	25%	100%	25%
3	1	1	25%	100%	25%

1	2
100%	50%

Thus the target person's coverage is the sum of the R values: 12.5% + 25% + 25% = 62.5%.

The green numbers are the pieces of the Venn diagram. They are the same as the three rows in the table.

- Child 1 (in brown) is the only child in piece 2. This is reflected in the table by a "1" in row 2 for Child 1 and a "0" in row 2 for Child 2.
- Child 2 (in red) is the only child in piece 1. This is reflected in the table by a "1" in row 1 for Child 2 and a "0" in row 1 for Child 1.
- Both children are in the center piece 3. This is reflected in the table by a "1" in row 3 for both Child 1 and Child 2.

So, each piece of the Venn diagram is duplicated in tabular form as a row in Table 1. Every part of the Venn diagram is in the table. Nothing is left out. The diagram and the table are the same by definition.

Table 2 is placed below Table 1.

- Child 1 tested so that 100% of that child's DNA is available to cover the parent.
- Child 2 did not test but has a child who did test. So, that tested grandchild covers 50% of Child 2.

This is what the target person's Table 2 has calculated by the time the algorithm's recursive calls returns to the top level just prior to the final calculations.

The final calculations take place row by row.

Row 1 is piece 1 of the Venn diagram – the piece unique to Child 2. (We see this with the 01 entries in the child columns for the row.)

- M is the maximum possible coverage of each piece of the Venn diagram. In this case of 2 children, M=25%. We see M both in the Venn diagram and in the table.
- W in row 1 (piece 1) can only be calculated from Child 2 since Child 1 is not in piece 1 (Child 1 has "0" in row 1.) If Child 2 had tested, then 100% of their DNA would be available for coverage of the target person. But Child 2 did not test. So, only 50% (half) of the DNA of Child 2 is available – through the test of the child of Child 2.
- So, instead of piece 1 supplying the full 25% maximum coverage, it provides only half of that 25%. So, R = 25% x 50% = 12.5% for row/piece 1.

Row 2 is piece 2 of the Venn diagram – the piece unique to Child 1. (In this case, it is Child 2 has a "0" in the row.)

- M is the same 25% as it is for every piece of the Venn diagram.
- W is now 100% since Child 1 did a DNA test.
- So, Child 1 can cover the entire 100% of piece 2. But piece 2 can only provide 25% of the coverage of the target person. So, R = 25% x 100% = 25%.

Row 3 is piece 3 of the Venn diagram – the piece unique that includes both Child 1 and Child 2. (In this case, both Child 1 and Child 2 have a "1" in the row.)

- M is the same 25% as it is for every piece of the Venn diagram.
- W is now 100% since Child 1's DNA test provides 100% of his/her possible coverage of the parent. Thus the 50% available from Child 2 is already part of what Child 1 can provide and effectively provides no added value. The algorithm uses the maximum value of all the children in the combination of multiple children, effectively providing a lower bound for the coverage that the children could provide for the parent. (In the scenario of Figure 4, the lower bound is the

same as the upper bound.) See Appendix 4 for a full discussion of combining the results of multiple children and how these really are parts of a range of values and not a single value when no child of the target person has tested.

- So, Child 1 can cover the entire 100% of piece 2. But piece 2 can only provide 25% of the coverage of the target person. So, R = 25% x 100% = 25%.

Writing the Actual Computer Code

The pseudo code is relatively simple – six steps with a two-step subroutine. The actual management of the tables and variable requires care but is well within the repertoire of any good programmer.

I do not specify the data structure of the input family tree. Different ways of handling the input already exist. Jonny Perl's DNA Painter WATO GUI front end for the specification of the input is a near-perfect fit for the envisioned tool. Similarly, David Stumpf's Graphs for Genealogists graph database fits very well as a front-end specification.

I intend this presentation of the method to allow anyone to implement the pseudo code in actual code.

Autosomal DNA Coverage of Past-Conflict Dead

It is a sad reality that the remains of many soldiers lost in World War II have never been recovered and identified. The U. S. military cemeteries have thousands of graves marked "Unknown", and the U. S. Defense POW-MIA Accounting Agency (DPAA) still recovers newly discovered World War II even this many years later.

DPAA obtains family DNA reference samples from members of the family of those men still unaccounted. Sometimes, the few surviving family members relate more distantly to the missing soldier than desired but are the only members of the family who can provide DNA.

I believe that a tool to calculate the percentage of coverage of the autosomal DNA of a still unaccounted service member could be useful for identifying currently unknown remains. This tool would be similar to the one described in this paper but would have the missing service member as the target person.

I have not yet attempted to create a detailed vision of this tool.

APPENDIX 1: Example with 3 Children

Here is an example of the relevant tables for a target person T with 3 children. This is the input specification.

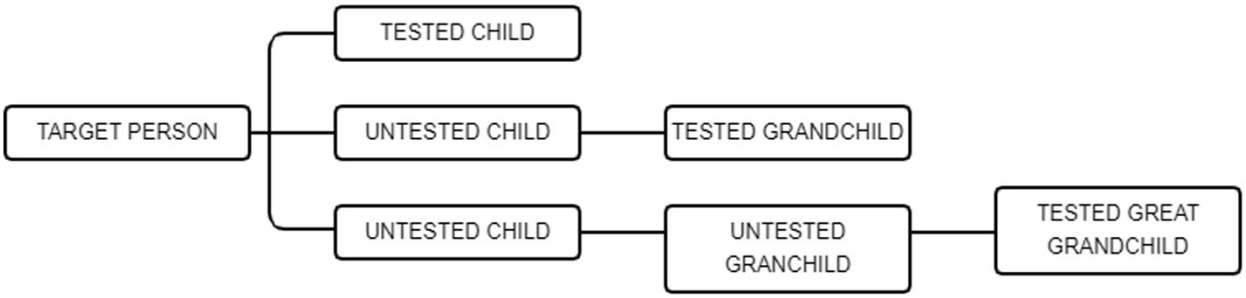


Figure 5-Example: 3 Children of Target Person

Call Level 1 – Reference Person = Target Person

The scope of each call is the reference person and his or her children. So, this level of the call looks only at these people. Each child then becomes the reference person in 2nd level calls which look at that child and his or her children. So, the scope of this first level call is only these people.

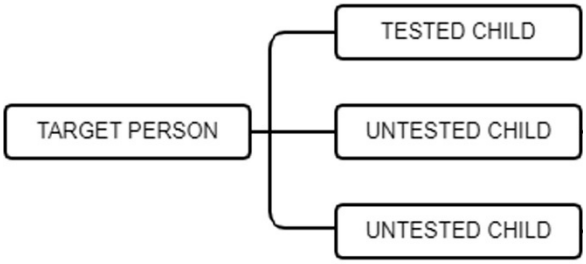


Figure 6-Call Level 1 Input Scope

1. Count the number of children of the reference person (in the first call, this is the target person). Set this as N. If N=0, return the coverage percentage as 100% and exit the module.
2. Calculate the number of pieces P of DNA (as in Figure 3) and the maximum coverage percentage M that each piece holds of the reference person's DNA.
3. Generate the module-internal tables, setting values to be calculated initially at zero.

Step 1 determines that **N=3**. So, it continues to step 2 which determines that **P = (2^N)-1 = 7** and that **M = 1/(2^N) = 1/8 = 0.125 = 12.5%**.

Step 3 generates the initial state of Tables 1 and 2.

Table 1 - People and Pieces
Initialized for call level 1

Piece\Child	1	2	3	M	W	R
1	0	0	1	12.5%	0%	0%
2	0	1	0	12.5%	0%	0%
3	0	1	1	12.5%	0%	0%
4	1	0	0	12.5%	0%	0%
5	1	0	1	12.5%	0%	0%
6	1	1	0	12.5%	0%	0%
7	1	1	1	12.5%	0%	0%

Table 2 – Tested/Reconstructed DNA of Children
Initialized for call level 1

1	2	3
0%	0%	0%

4. For each child, determine the coverage of the child by calling this same module.

This level of the module will now issue a separate call for each of the three children.

Call Level 2 – Reference Person = Child 1 of Target Person

Since child 1 has no children, the scope of this call is only child 1 (the teste child) of the target person.

1. Count the number of children of the reference person (in the first call, this is the target person). Set this as N. If N=0, return the coverage percentage as 100% and exit the module.

Since child 1 has no children, N = 0, and this call returns a value of 100% coverage and exits.

Resume Call Level 1 – Reference Person = Target Person

The call for child 1 returned 100%. So, Table 2 now has these values.

Table 2 – Tested/Reconstructed DNA of Children
Call level 1 after level 2 call for child 1

1	2	3
100%	0%	0%

4. For each child, determine the coverage of the child by calling this same module.

The iteration through step 4 continues with a call of the module for child 2.

Call Level 2 – Reference Person = Child 1 of Target Person

Since child 1 has no children, the scope of this call is only child 1 (the teste child) of the target person.

Figure 7-Call Level 2 for Child 2 as the Reference Person

1. Count the number of children of the reference person (in the first call, this is the target person). Set this as N. If N=0, return the coverage percentage as 100% and exit the module.
2. Calculate the number of pieces P of DNA (as in Figure 3) and the maximum coverage percentage M that each piece holds of the reference person's DNA.
3. Generate the module-internal tables, setting values to be calculated initially at zero.

Step 1 determines that **N=1**. So, it continues to step 2 which determines that **P = (2^N)-1 = 1** and that **M = 1/(2^N) = 1/2 = 0.5 = 50%**.

Step 3 generates the initial state of Tables 1 and 2.

Table 1 - People and Pieces
Initialized for call level 2 for child 2 of the target person

Piece\Child	**1**	**M**	**W**	**R**
1	1	50%	0%	0%

Table 2 – Tested/Reconstructed DNA of Children
Initialized for call level 2 for child 2 of the target person

1
0%

4. For each child, determine the coverage of the child by calling this same module.

This level of the module will now issue a separate call for the lone child of child 2 of the target person.

The iteration through step 4 continues with a call of the module for child 2.

Call Level 3 – Reference Person = Child of Child 1 of Target Person

Since child 1 has no children, the scope of this call is only child 1 (the teste child) of the target person.

Figure 8-Call Level 3 for Child of Child 2 as the Reference Person

1. Count the number of children of the reference person (in the first call, this is the target person). Set this as N. If N=0, return the coverage percentage as 100% and exit the module.

Since the tested child has no children, N = 0, and this call returns a value of 100% coverage and exits.

Resume Call Level 2 – Reference Person = Child 2 of Target Person

The call for the child returned 100%. So, Table 2 now has these values.

Table 2 – Tested/Reconstructed DNA of Children
Call level 2 for child 2 of the target person after level 3 call of that person's child

1
100%

5. For each piece/row of Table 1, calculate W and R for that piece/row.

Here is the initial state of Table 1 for this call.

Table 1 - People and Pieces
Initialized for call level 2 for child 2 of the target person

Piece\Child	1	M	W	R
1	1	50%	0%	0%

Step 5 enters the returned value (100%) as the value of W(1) since cell(1,1) =1 so that W = 1 * 100%.

Table 1 - People and Pieces
Weight calculated for call level 2 for child 2 of the target person

Piece\Child	1	M	W	R
1	1	50%	100%	0%

Step 5 then computes R = M * W(1) = 50% * 100% = 50%

Table 1 - People and Pieces
Weight calculated for call level 2 for child 2 of the target person

Piece\Child	1	M	W	R
1	1	50%	100%	50%

6. Add the percentages for the pieces together to calculate the coverage percentage of the reference person and return that number and exit the module.

Since there is only one piece/row, the sum is 50%. This call returns 50% and exits.

Resume Call Level 1 – Reference Person = Target Person

Table 1 remains the same. Table 2 now has the value for child 2.

Table 1 - People and Pieces
Initialized for call level 1

Piece\Child	1	2	3	M	W	R
1	0	0	1	12.5%	0%	0%
2	0	1	0	12.5%	0%	0%
3	0	1	1	12.5%	0%	0%
4	1	0	0	12.5%	0%	0%
5	1	0	1	12.5%	0%	0%
6	1	1	0	12.5%	0%	0%
7	1	1	1	12.5%	0%	0%

Table 2 – Tested/Reconstructed DNA of Children
Call level 1 after level 2 call for child 2

1	2	3
100%	50%	0%

4. For each child, determine the coverage of the child by calling this same module.

This level of the module will now issue a separate call child 3 of the target person.

The iteration through step 4 continues with a call of the module for child 3.

Call Level 2 – Reference Person = Child 3 of Target Person

I omit the detail of this calculation since it is now clear from the call for child 2 how the nested calls will work for child 3 to ultimately return the value 25% from this call for the third child.

Resume Call Level 1 – Reference Person = Target Person

Table 1 remains the same. Table 2 now has the value for child 3.

Table 1 - People and Pieces
Initialized for call level 1

Piece\Child	1	2	3	M	W	R
1	0	0	1	12.5%	0%	0%
2	0	1	0	12.5%	0%	0%
3	0	1	1	12.5%	0%	0%
4	1	0	0	12.5%	0%	0%
5	1	0	1	12.5%	0%	0%
6	1	1	0	12.5%	0%	0%
7	1	1	1	12.5%	0%	0%

Table 2 – Tested/Reconstructed DNA of Children
Call level 1 after level 2 call for child 2

1	2	3
100%	50%	25%

5. For each piece/row of Table 1, calculate W and R for that piece/row.

Row 1 Calculation

W(1) is the maximal value of these three (0 * 100%, 0 * 50%, 1 * 25%) = (0, 0, 25%). The zeros and ones are from the child columns of row 1 of Table 1. The percent values are from the corresponding cells of Table 2. So, W(1) = 25%, since that is the maximum value of the three products.

Then R(1) = M * W(1) = 12.5% * 25% = 3.125%. So, Table 1 now has these values.

Table 1 - People and Pieces
Call level 1 after calculation of W(1) and R(1)

Piece\Child	1	2	3	M	W	R
1	0	0	1	12.5%	25%	3.125%
2	0	1	0	12.5%	0%	0%
3	0	1	1	12.5%	0%	0%
4	1	0	0	12.5%	0%	0%
5	1	0	1	12.5%	0%	0%
6	1	1	0	12.5%	0%	0%
7	1	1	1	12.5%	0%	0%

Row 3 Calculation

Row 2 is a single-child piece so that its calculation follows a similar path as row 1, with the only difference that child 2 and not child 3 determines the values of W(2) and R(2). Row 3 differs in that it is the first row (the first piece of the Venn diagram) with a combination of children.

W(3) is the maximal value of these three (0 * 100%, 1 * 50%, 1 * 25%) = (0, 50%, 25%). The zeros and ones are from the child columns of row 3 of Table 1. The percent values are from the corresponding cells of Table 2. So, W(3) = 50%, since that is the maximum value of the three products.

Then R(3) = M * W(3) = 12.5% * 50% = 6.25%. So, Table 1 now has these values.

Table 1 - People and Pieces
Call level 1 after calculation of W(3) and R(3)

Piece\Child	1	2	3	M	W	R
1	0	0	1	12.5%	25%	3.125%
2	0	1	0	12.5%	50%	6.25%
3	0	1	1	12.5%	50%	6.25%
4	1	0	0	12.5%	0%	0%
5	1	0	1	12.5%	0%	0%
6	1	1	0	12.5%	0%	0%
7	1	1	1	12.5%	0%	0%

Row 7 Calculation

The calculation of rows 4-6 follows a similar path as rows 1-3. Row 73 differs in that it is the first row (the first piece of the Venn diagram) with a combination of all three children.

W(7) is the maximal value of these three (1 * 100%, 1 * 50%, 1 * 25%) = (100%, 50%, 25%). The zeros and ones are from the child columns of row 7 of Table 1. The percent values are from the corresponding cells of Table 2. So, W(7) = 100%, since that is the maximum value of the three products.

Then R(7) = M * W(7) = 12.5% * 100% = 12.5%. So, the final state of Table 1 now has these values.

Table 1 - People and Pieces
Call level 1 after full calculation

Piece\Child	1	2	3	M	W	R
1	0	0	1	12.5%	25%	3.125%
2	0	1	0	12.5%	50%	6.25%
3	0	1	1	12.5%	50%	6.25%
4	1	0	0	12.5%	100%	12.5%
5	1	0	1	12.5%	100%	12.5%
6	1	1	0	12.5%	100%	12.5%
7	1	1	1	12.5%	100%	12.5%

6. Add the percentages for the pieces together to calculate the coverage percentage of the reference person and return that number and exit the module.

The module ends by summing up the values of the pieces and returning that sum and exiting.

The sum is 3.125% + 6.25% + 6.25% + 12.5% + 12.5% + 12.5% + 12.5% = 65.625%.

APPENDIX 2: Differing Input Order

The order in which the input tree presents the children does not alter the output. Here are three examples of the final state of Table 1 and Table 2 in the top-level call of the module just before it sums up the pieces of the Venn diagram, outputs that sum and exits. In all cases, the output is the same.

In each case, the children are numbered from 1 at the top to 3 at the bottom.

Order of the Example in Appendix 1

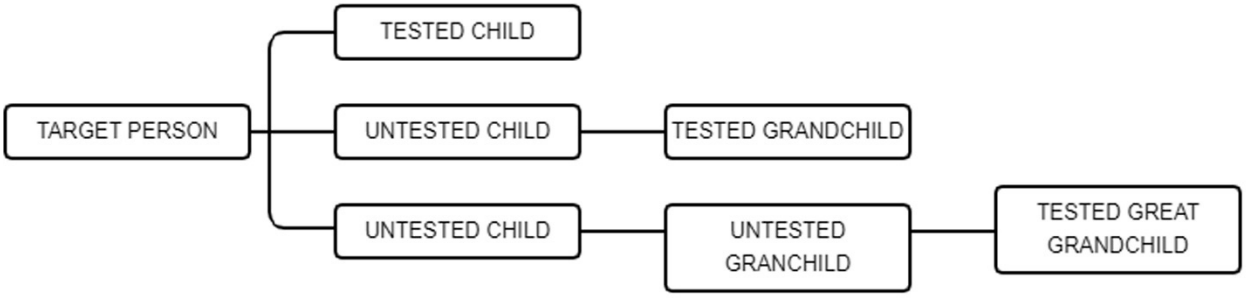


Table 2 – Tested/Reconstructed DNA of Children
Final Call level 1 in example in Appendix 1

1	2	3
100%	50%	25%

Table 1 - People and Pieces
Final Call level 1 in example in Appendix 1

Piece\Child	1	2	3	M	W	R
1	0	0	1	12.5%	25%	3.125%
2	0	1	0	12.5%	50%	6.25%
3	0	1	1	12.5%	50%	6.25%
4	1	0	0	12.5%	100%	12.5%
5	1	0	1	12.5%	100%	12.5%
6	1	1	0	12.5%	100%	12.5%
7	1	1	1	12.5%	100%	12.5%

The module returns the sum of column R values = 65.625%.

Alternate Input Order 1

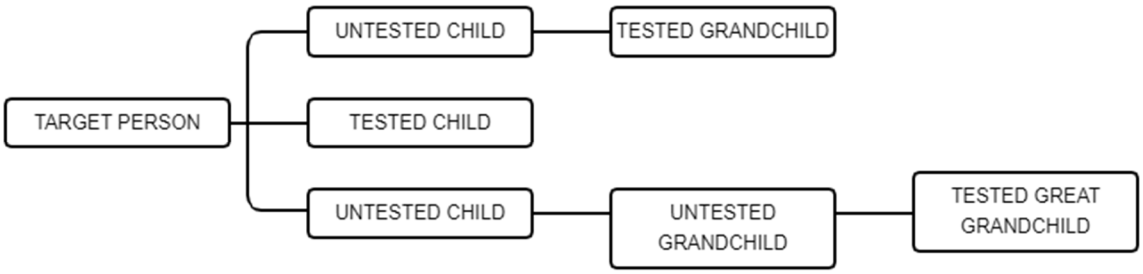


Table 2 – Tested/Reconstructed DNA of Children
Final Call level 1 in Alternate Input Order 1

1	2	3
50%	100%	25%

Table 1 - People and Pieces
Final Call level 1 in Alternate Input Order 1

Piece\Child	1	2	3	M	W	R
1	0	0	1	12.5%	25%	3.125%
2	0	1	0	12.5%	100%	12.5%
3	0	1	1	12.5%	100%	12.5%
4	1	0	0	12.5%	50%	6.25%
5	1	0	1	12.5%	50%	6.25%
6	1	1	0	12.5%	100%	12.5%
7	1	1	1	12.5%	100%	12.5%

The module returns the sum of column R values = 65.625%.

Alternate Input Order 2

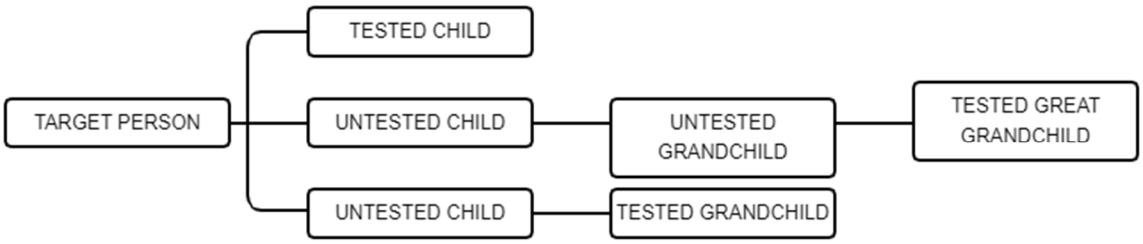


Table 2 – Tested/Reconstructed DNA of Children
Final Call level 1 in Alternate Input Order 2

1	2	3
100%	25%	50%

Table 1 - People and Pieces
Final Call level 1 in Alternate Input Order 2

Piece\Child	1	2	3	M	W	R
1	0	0	1	12.5%	50%	6.25%
2	0	1	0	12.5%	25%	3.125%
3	0	1	1	12.5%	50%	6.25%
4	1	0	0	12.5%	100%	12.5%
5	1	0	1	12.5%	100%	12.5%
6	1	1	0	12.5%	100%	12.5%
7	1	1	1	12.5%	100%	12.5%

The module returns the sum of column R values = 65.625%.

APPENDIX 3: Pedigree Collapse

While the DNA Painter WATO graphical user interface does not allow specification of pedigree collapse scenarios, they did happen, and they do impact how the DNA tests of the descendants cover the ancestor. The method described in the main part of this paper handles such cases.

Take the relatively simple case of two first cousins who married. The target person is one of their parents. The test taker is one of their children. If WATO could handle the specification of this scenario, it might look like this.

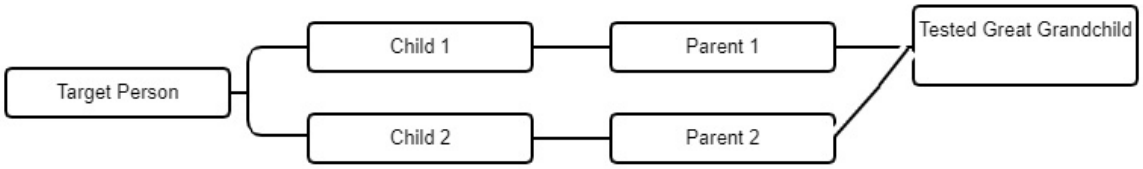


Figure 9-Pedigree Collapse - Simple Example Input Specification

Because the recursive calls work down from the target person for each child, the tested descendant provides the DNA for two different children in this scenario. Here are the two separate paths.

Target Person – Child 1 – Parent 1 – Tested Grandchild

Target Person – Child 2 – Parent 2 – Tested Grandchild

The tested grandchild provides 50% coverage for Parent 1 and 25% coverage for Child 1. The same tested grandchild provides 50% coverage for Parent 2 and 25% coverage for Child 2. So, we have the final table values for the target person as follows.

Table 2 – Tested/Reconstructed DNA of Children

1	2
25%	25%

Table 1 - People and Pieces

Piece\Child	1	2	3	M	W	R
1	0	0	1	25%	25%	6.25%
2	0	1	0	25%	25%	6.25%
3	0	1	1	25%	25%	6.25%

The module returns the sum of column R values = 18.75%.

Journal of Genetic Genealogy

Appendix 4: Calculating the Shared DNA Component: Dealing with Ranges

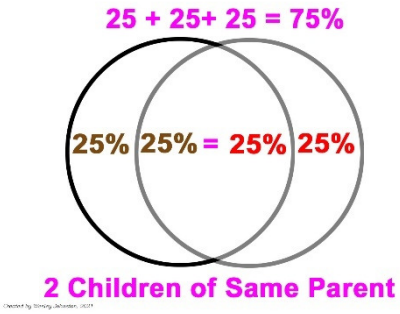
The calculation of a parent's DNA coverage from the DNA of tested descendants requires dealing with the DNA unique to each child and also to the combination of DNA from multiple children. A single number represents the amount of DNA a child can contribute to a parent in some cases. But when the coverage comes from the combination of testers who are grandchildren or more distant descendants of the ancestor, it is impossible in a theoretical estimate of coverage to know precisely which regions of DNA those children's descendants provide in their coverage. The reality is that a range of possible values results from the combination. There is no way in a theoretical model to tell just which case represents the DNA that each child contributes in a specific actual family.

This appendix explains in detail how this reality impacts any attempt to calculate theoretical coverage estimates.

The algorithm deals with this reality by calculating the lower bound of each range. The resulting coverage value is very likely less than what the actual coverage of the target person would be if the precise DNA being used in an actual case was known. But as the lower bound, it assures that the estimated coverage is at least that high.[iv]

The Two-Child Scenario

Two children who have both autosomal DNA tested provide coverage for their designated parent as shown in this diagram.



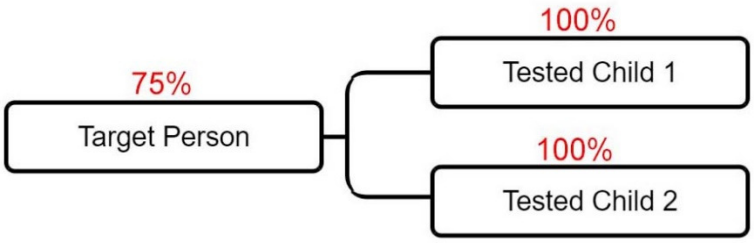
The left circle indicates the DNA of Child 1. The overlapping right circle indicates the DNA of Child 2. The arc (with the brown 25%) on the left side is DNA of the parent that is unique to Child 1. The arc (with the red 25%) on the right side is DNA of the parent that is unique to Child 2. The center area (with the brown and red 25%=25%) is DNA of the parent that both children share identically.

Each child inherited half of their DNA from each parent. Thus, each child has half of their DNA from the designated parent. Roughly half of each child's DNA inherited from the designated parent (thus 25% of the designated parent's DNA) is unique to that child, and the other half (thus also 25% of the parent's DNA) is identical to DNA inherited by the other child.

The diagram shows the unique 25% for Child 1 as the Brown 25% on the left, outside the center area. And it shows the unique 25% for Child 2 as the red 25% on the right, outside the center area.

The center area shows the identical 25% of the parent's DNA inherited by both children.

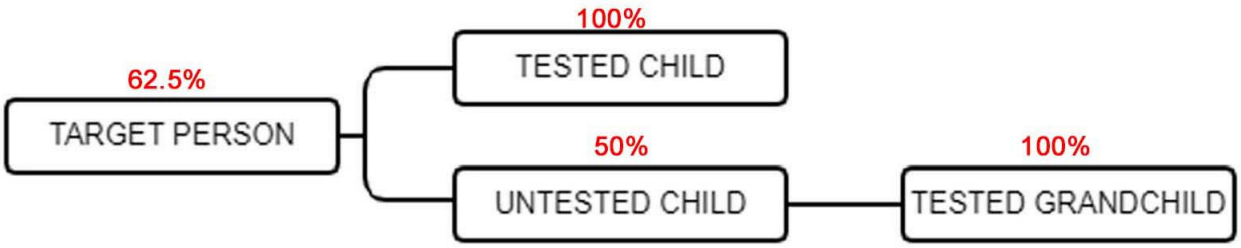
This diagram represents the scenario of the parent and two children in Jonny Perl's DNA Painter WATO graphical user interface format.



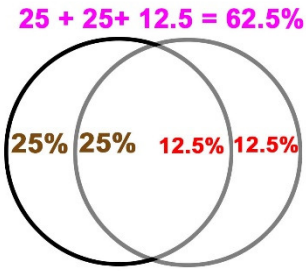
The two children combine to cover 75% of the parent: 25% unique from each child plus the identical 25% that they share = 25% + 25% + 25% just as the top diagram shows.

The One-Child One-Grandchild Scenario

The first extension of the 2-child scenario is when one child tests along with a child of the other child. This diagram represents the scenario in WATO graphical user interface format.



The Venn diagram represents the scenario as follows.



The three pieces each contribute to the overall total as follows.

Child 1 Unique Piece (left arc) = 25%

Child 2 Unique Piece (left arc) = 12.5%

Center Shared Piece = 25%

The crucial aspect of this calculation is how the center shared piece value is calculated when the two children do not have identical amounts of shared DNA.

Only half of Child 2's DNA was also inherited by Child 1. That reality does not change. Child 1 has inherited 25% of the parent's DNA that would have been identical to Child 2's DNA if Child 2 had tested. That reality has not changed from the first scenario. What has changed is that Child 2's DNA is only 12.5% in this scenario.

Child 1 has already provided the full 25% that would have been identical if Child 2 had tested. So, it really does not matter what percent Child 2 provides in the center shared area. We already have as much from child 1 as we can possibly have. Child 2 cannot add any more to the 25% since that is the maximum possible amount that they share.

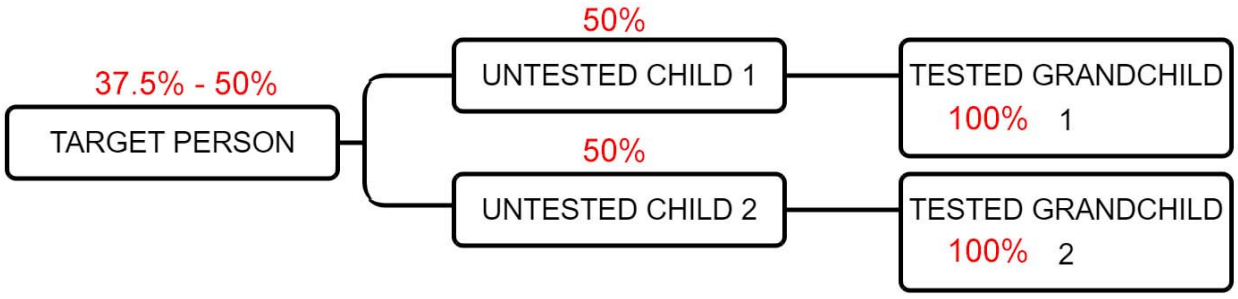
The result is that the maximum value in the center area is in fact the amount of coverage that this shared piece of DNA can provide to the parent. In no case can it be more than 25%.

So, the center shared piece value is simply the larger (the maximum) of the two values available for coverage of the parent. In this case of 25% and 12.5%, the larger (maximum) value is 25%.

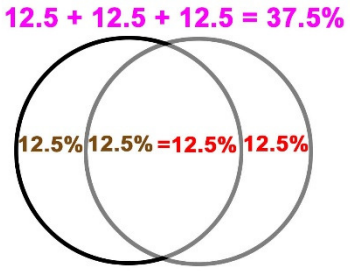
The total coverage is the sum of the three pieces: 25% + 12.5% + 25% = 62.5%.

The Two-Grandchildren Scenario: All children < 100% (Range)

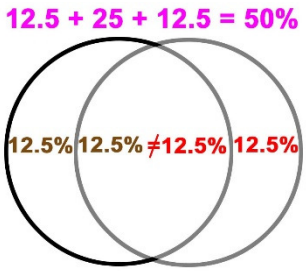
The next extension of the 2-child scenario is when neither child tests but a child of each child tests. This diagram represents the scenario in WATO graphical user interface format.



This Venn diagram represents one case (where the shared DNA is identical).



This Venn diagram represents another case (where the shared DNA from each child does not match any of the shared DNA from the other child).



The three pieces each contribute to the overall total as follows.

Child 1 Unique Piece (left arc) = 12.5%

Child 2 Unique Piece (left arc) = 12.5%

Center Shared Piece = 12.5% to 25% -- a range and not a single number

Having no children of the target person who have tested represents the actual situation for most parent-child scenarios as a distant relative's coverage is calculated. And it presents a radically different reality than was the case for the two previous scenarios. The radical change results from the fact that none of the children provide full coverage (25%) of this piece of the Venn diagram. The result is that the coverage is really a range of values and not a single number.

In this scenario, one case may be as in the first Venn diagram above: the 12.5% provided by Child 1 is identical to the 12.5% provided by Child 2. In this case, the value of the center shared area is 12.5% so that the total of the three pieces is 37.5% (12.5% + 12.5% + 12.5%), as shown in the first Venn diagram above.

But it is more likely that some or all of the DNA that Child 1 received from the parent did not pass down to their own child. And the same is true for Child 2. It could be that the 12.5% provided by the reconstructed DNA of Child 1 does not match any of the 12.5% reconstructed DNA of Child 2. In that case, the center shared area has a value of the full 25% possible (12.5% + 12.5%) so that the total of the three pieces is 50% (12.5% + 25% + 12.5%), as shown in the second Venn diagram above.

In fact, the actual situation is that the center shared area can have any value from 12.5% to 25%, depending on how much of the reconstructed DNA of Child 1 is identical (or not) to the reconstructed DNA of Child 2. I have not done the calculations on this, but I believe that the possible combinations form a normal distribution with 12.5% at the low end and 25% at the high end and the most frequent cluster of possible combinations at the midpoint of 18.75% which would result in a total coverage of the parent as 43.75% (which is the same as the total coverage for this scenario using Paul Woodbury's formula).[v]

The reality is that there is no single number that represents the center shared area because any combination of the two children's reconstructed DNA contribution to the center shared area can happen. The center shaded area is a range from 12.5% to 25% so that the total coverage is a range from 37.5% to 50%. So, any theoretical model resulting in a single number has to choose one number from within the range. But to most accurately reflect the situation, a range and not a single number is the actual result.

For this scenario, here are the calculations for three key numbers that can be used to represent the situation in a single number instead of as a range.

Lower Bound = 37.5%

Choosing the maximum of the two percentages available results in the total coverage estimate being a lower bound: the kits of the descendants will cover at least this much of the target person's DNA. Choosing the maximum percentage effectively chooses the case in which the two children's reconstructed DNA is identical. This choice easily fits within the generalized recursive algorithm.

Upper Bound = 50%

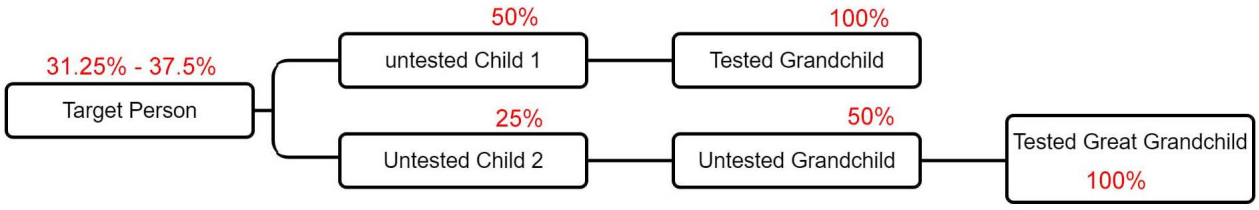
The upper bound is the sum of the percentages from each child within this piece of the Venn diagram. In no case can it be larger than the size of the piece of the diagram (the value of M in the algorithm's Table 1). So, the value is the minimum of the piece size (25% in the case of 2 children in this scenario) or the sum of the children's percentages for the center shared area.

Most Frequent Combination (Mean or Average) = 43.75%

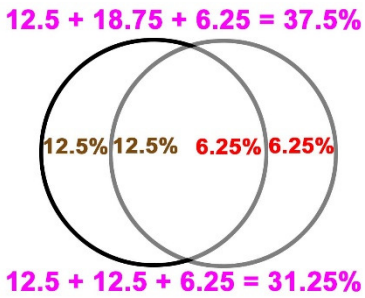
This is simply the mean of the low end (the maximum single percentage) and the high end (the sum of the percentages of the children). Since the mean cannot be greater than the size of the piece of the Venn diagram, the actual value used must be the minimum of the piece size (25% in this scenario) and the calculated mean.

The One-Grandchild One-Great Grandchild Scenario

The next extension of the 2-child scenario is when neither child tests but a child of one child tests and a grandchild of the other child tests. This diagram represents the scenario in WATO graphical user interface format.



The Venn diagram for this scenario looks like this, showing both the lower (bottom) and upper (top) bounds.

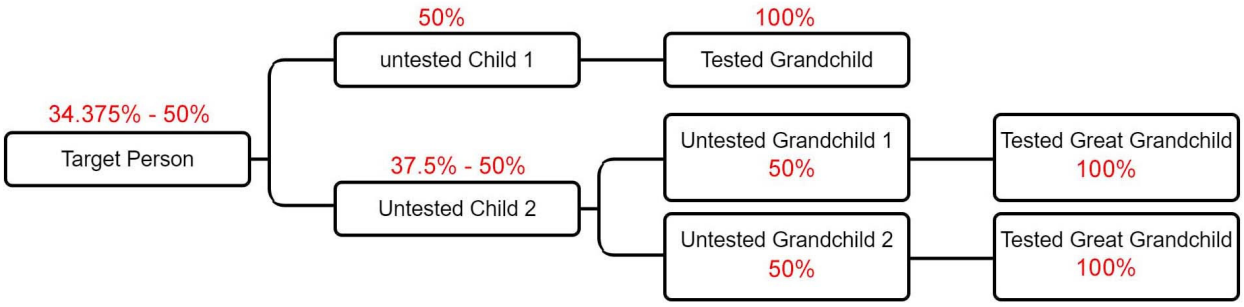


The algorithm returns 31.25%, which is the lower bound.

The average of the total coverage based on the upper and lower bounds is 34.375. (This is the same as the value calculated by Paul Woodbury's formula.)

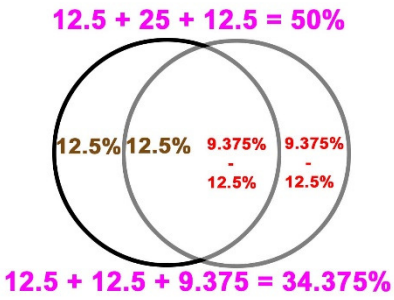
The One-Grandchild Two-Great Grandchildren Scenario (2 Ranges)

The next extension of the 2-child scenario is when neither child tests but a child of one child tests and a grandchild of the other child tests. This diagram represents the scenario in WATO graphical user interface format.



This is the first scenario in which we have had to deal with a range being passed up to a target person. If Child 1 had tested, the range would not matter since Child 1 could provide the full center shared area. But Child 1 has not tested. So, the range does require inclusion in the final calculation of the target person's coverage – a calculation which as we saw in the prior scenarios also results in a range and not a single number.

The Venn diagram for the target person in this scenario looks like this.



The diagram shows the lower bound of the target ancestor's coverage at the bottom and the upper bound at the top. So, the actual result is the range 34.375% (which is what the algorithm will return) to 50%.

The algorithm reconstructs the coverage for Child 2 using the two grandchildren's 50%. Thus, the algorithm calculates Child 2 as 37.5%. So, the final coverage by the algorithm sees Child 1 (50%) and Child 2 (37.5%) and calculates the parent as 34.375% which is the same as the lower bound for the range.

The average of the upper and lower bounds for the target person's coverage range is 42.1875%. Woodbury's formula calculates to 41.40625%. The difference is due to the formula passing forward only

a single number for Child 2 and not the range. The upper bound for both the target person and Child 2 remains at 50%, but the lower bound drops from 37.5% for Child 2 to 34.375% for the target person. Thus, the average for Child 2 within the range is the same as for the formula. But because only the lower bound changed between Child 2 and the target person, the formula's use of the average from Child 2 in the calculation of the total coverage of the target person diverges from the average of the range for the target person. This is because of the negative term in this paradigm, subtracting the product of half of the two children's available DNA, results in a subtraction in the coverage of the target person of more than the amount that reaches the true halfway point of the range and thus results in a calculated value below the true midpoint/average of the range.

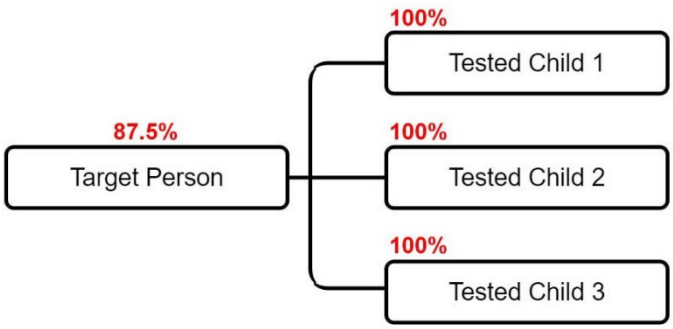
In other words, on the normal distribution curve of the possible combinations of cases for the target person's range, the average of that range is the most probable case, but the formula results in a value that is a slightly less probable case. This divergence of the formula from the true average of the range (the most probable combination of cases within the range) increases as more ranges are combined as steps in the calculation of the coverage of the target person.

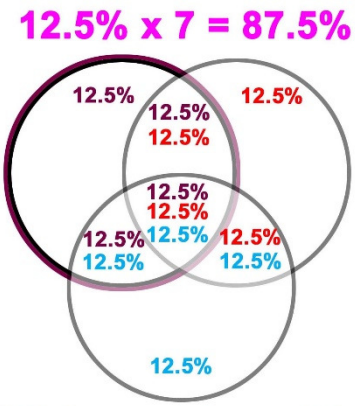
This skewing is not the case if the lower bound or the upper bound or the average of the lower and upper bounds is used as the single number for the coverage of each person. The real issue is that the true coverage is a range and not a single number. So, if the average (no matter how it is calculated) is used as the single number, then it will diverge from the true average. If the average is used, then the full calculation of the bounds and thus the range has to be used at every step where reconstructed DNA is less than 100% from at least one child.

Correctly carrying forward (from child to parent in the calculation) the full range requires a different formula and a different algorithm. The algorithm in this paper does provide a stable and consistent single number (the lower bound) for each step so that it does not skew due to repeated steps passing forward a single number instead of the full range. Even if the full range was passed forward, the lower bound would be the same. The algorithm would be more accurate if it could deal with the combinations of the ranges and not just the lower bounds. Nevertheless, the algorithm is accurate in its calculation of a lower bound for the target person while the use of the average as the single number for a person skews lower than the true average as more ranges are combined.

The Three Child Scenario

The 3-child scenario adds complexity. The scenario and Venn diagram for three tested children look like



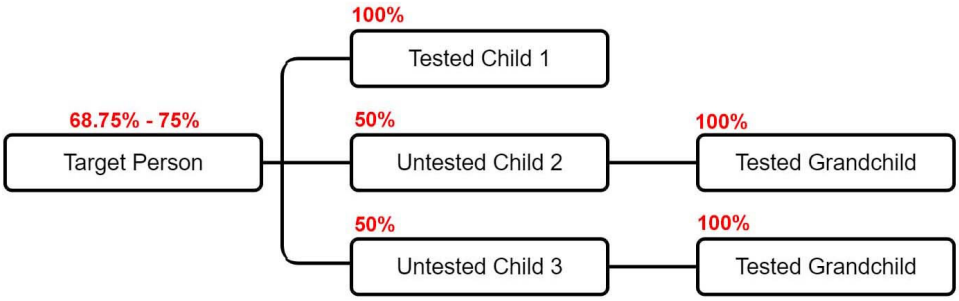


The Venn diagram now has 7 pieces, each with a maximum of 12.5% that it can contribute to the total coverage of the parent. Instead of one paired combination, the diagram has three paired combinations (Children 1 and 2, 1 and 3, 2 and 3). And the center shared area now combines the contributions of all three children.

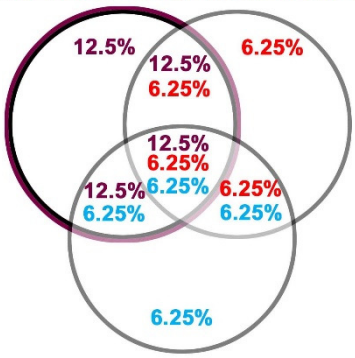
In this scenario, all three children tested. So, the coverage of the parent is the sum of all seven parts, each of which contributes its full 12.5% to the parent's coverage.

The One-Child Plus Two-Grandchildren Scenario (Range)

This scenario provides a full contribution (12.5%) in the shared center area and for two of the paired combinations since one child did test. So, these pieces of the Venn diagram provide a single number (12.5%). But one of the three paired combinations has only reconstructed DNA for the children.



12.5%+6.25%+6.25%+12.5%+12.5%+12.5%+12.5%=75%
12.5%+6.25%+6.25%+12.5%+12.5%+6.25%+12.5%=68.75%

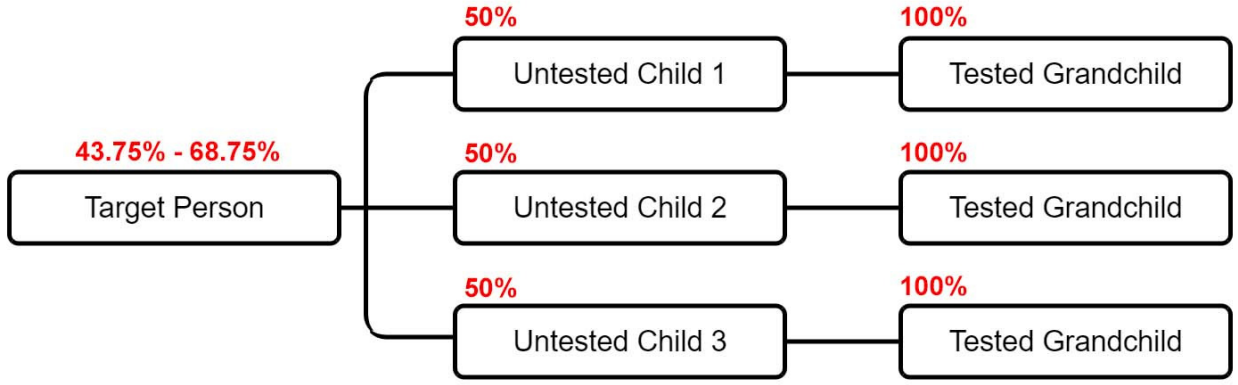


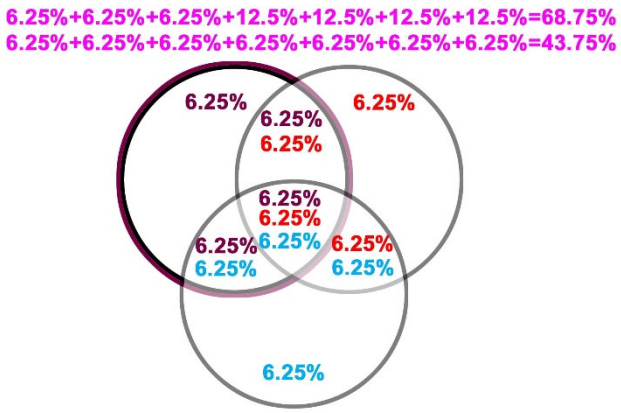
The Venn diagram shows the upper (75%) and lower bound (68.75%) calculations above the overlapping circles. The intersection of Child 2 (red) and Child 3 (blue) results in a range and not a single number in the same way that we have seen in the 2-child scenarios.

The average of the upper and lower bounds is 71.875%, which is the value calculated by Paul Woodbury's formula.

The Three-Grandchildren Scenario: All children < 100% (Range)

This scenario has no full contribution (12.5%) in the shared areas. So, all of the shared areas result in ranges.





The Venn diagram shows the upper (68.75%) and lower bound (43.75%) calculations above the overlapping circles. The inclusion of more and more ranges results in a wide total coverage range. The upper bound exceeds the lower bound by 25%.

The average of the upper and lower bounds is 56.25%. Paul Woodbury's formula calculates total coverage of the target person as 57.8125%. As with the multi-step range cases of the 2-child scenarios, the formula is close to the average but not quite the same. In this case, the formula results in a value greater than the actual average.

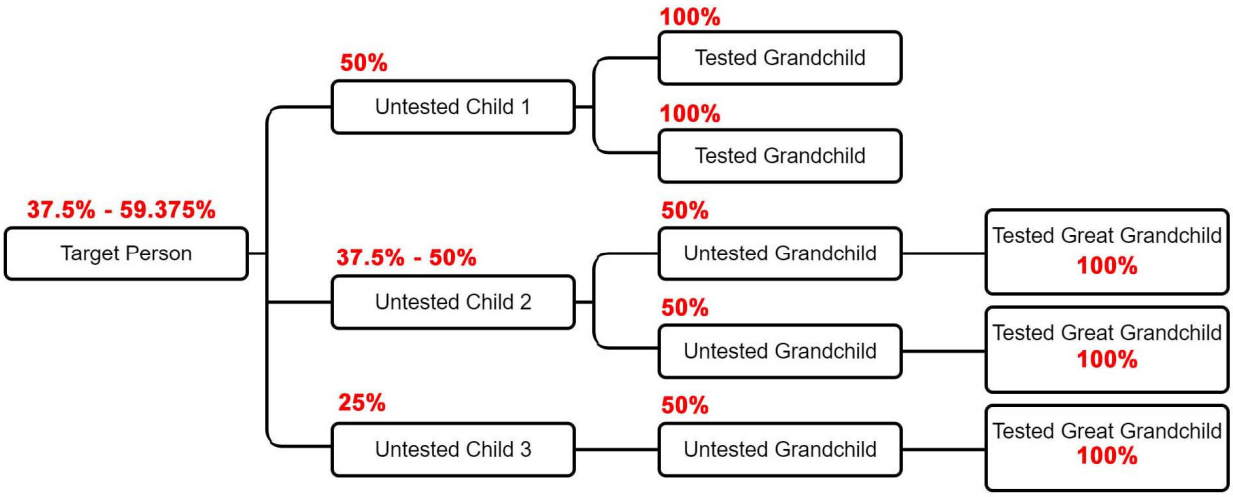
Note that the calculation of the contribution of the center shaded area runs into the upper limit of 12.5% for any one piece of the Venn diagram. Adding the two 6.25% partial coverage values together for 2 children worked because in the 2-child case the center shaded area contributes up to 25% so that the 12.5% sum did not exceed the maximum. But in this scenario, adding the three children together gives

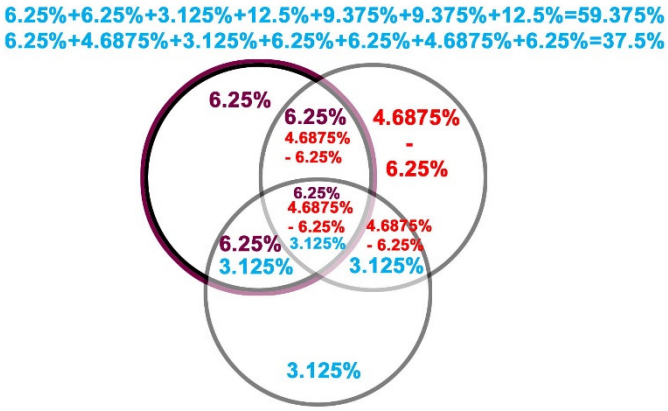
18.75% which is greater than the 12.5% this piece of the Venn diagram can contribute to the total coverage of the target person. Once Child 1 and Child 2 have covered the full 12.5% for the piece, the value for Child 3 does not alter the result. The three children can combine to supply at most 12.5% in the center shared piece, and they have done so in the upper bound case.

Since the algorithm calculates the lower bound (by using the maximum function), it returns 6.25% for each combination piece and thus a total coverage of 43.75% for the target person.

Complex Three-Grandchildren Scenario: All children < 100% (Range)

This scenario represents an input configuration that may well appear in a family. The scenario results in multiple range results to combine as they pass up from the test takers to the target person.





The average of the upper (59.375%) and lower (37.5%) bounds is 48.4375%. Paul Woodbury's formula calculates to 48.7305%.

The algorithm calculates Child 2 as 37.5%. For the final calculation for the target person's coverage, the algorithm sees Child 1 (50%), Child 2 (37.5%) and Child 3 (25%). It then returns the result 37.5% for the target person since all four combination pieces have a maximum weight of 50% (from Child 1).

Implications

The reality is calculations of the coverage of a person who has no children who have tested result in a range and not a single number.

The algorithm addresses this by always returning the lower bound of the range which does accurately pass forward from one generation to the next.

Paul Woodbury's formula addresses this by returning the average. This is accurate for cases in which at least one child has tested but diverges from the average of the range when the testers are more distant than children of the target person. Nevertheless, it is close to the average even in cases where multiple ranges of coverage combine.

Perhaps the ideal algorithm would generate the accurate upper and lower bounds of the ranges and their average in the final range so that all three numbers could be returned. For now, the algorithm returns only the lower bound.

---

[i] The key word here is "theoretical". The coverage is an estimate, based on the assumption that roughly 50% of a person's DNA comes from each parent. When none of the children of the target person (the person whose coverage is being calculated) have DNA-tested, the coverage estimate is not a single percentage but a range of possible percentages. This is due to the grandchildren and more distant descendants inheriting only part of the target person's DNA. In a theoretical model, we cannot know the exact percent and thus must either show it as a range or choose either the upper bound, the lower bound or the average of the two to represent a range as a single number (very much as ethnicity results shown as single percentages really represent a range).

There are further complications, such as noted in the work of Briton Nicholson creating relationship predictors that "take into account differences between maternal and paternal relationships". (See his 27 Jun 2018 "How Much of an Ancestor's DNA Do You Have?" https://beanmclellan.medium.com/how-much-of-an-ancestors-dna-do-you-have-b6959178471f "It happens that recombination from mothers to children is greater than that from fathers, resulting in more variability in lines that are majority male and less variability in lines that are majority female.")

Also, as cited in Appendix 4, Kevin Borland in 2019 explained why actual reconstruction of a DNA kit of an ancestor will not result in as much DNA as the theoretical coverage calculations.

So, while coverage calculation is not at all an exact science and can provide only theoretical estimates, nevertheless within the assumptions used for the theoretical model, it can be precisely calculated. Actual specific cases most likely will not match any single number prediction but will probably fall within the range from the lower bound to the upper bound.

The algorithm uses the lower bound of the range. It is accurately calculated in all scenarios and closer to what Kevin Borland indicates is likely to be possible in actual reconstruction of a fabricated DNA kit of an ancestor.

Paul Woodbury's formula calculates the average. Although the formula is difficult to scale and generalize and although it does not calculate the actual average once ranges and not single values of coverage are the reality, it does calculate the average in single-valued cases accurately and comes very close to the average in cases where ranges and not single values are required.

[ii] As coined by Paul Woodbury, coverage means the percentage of an ancestor's DNA that theoretically could be reconstructed from the test results of his or her descendants. I use it here in the sense of the percentage of the matches of an untested ancestor who his or her combined descendants' tests would match. Just as in the case of ranges of DNA coverage where an actual case will probably not match the theoretical upper or lower bound nor the average of the two, there are cases – such as excess IBD pileup regions among other reasons – where a kit may

actually have far more or fewer matches than the theoretical model predicts. But in the aggregate, the model is usable as a predictor of matches just as in the aggregate it is accurate as a predictor of theoretical DNA reconstruction.

[iii] Paul Woodbury, Briton Nicholson and Amy Williams all began independently working on this about the same time in 2016-2018. Briton Nicholson and Amy Williams have online tools and papers well worth reading. But it was Paul Woodbury who coined the term "coverage" and gave several presentations with a very good visual image of what coverage means. He first published on coverage in February 2017 as part of the blog: https://www.legacytree.com/blog/dna-testing-older-relatives-now

[iv] Kevin Borland whose Borland Genetics software reconstructs a fabricated DNA kit from those of descendants cautioned in 2019 posts on the Borland Genetics Facebook page that the actual reconstruction – especially in cases with unphased test results – will most likely reconstruct less than a theoretical estimate. So, the lower bound also has the effect of being more realistic about what can actually be reconstructed. (See https://www.facebook.com/groups/borlandgenetics/posts/404094730131696)

[v] Note that Paul Woodbury's formula operates with a paradigm applied to the Venn diagram that differs from the paradigm used here. Here, the paradigm is that the three separate pieces of the Venn diagram are calculated and then added together. This enables the generalization and scaling of the simple algorithm. Paul Woodbury's formula operates from a different paradigm in which the combined areas (the areas in which DNA is shared among the children) are subtracted from or added to the sum of the DNA available from all of the children.

The two paradigms work the same. The only difference is in which number to choose within the range of possibilities. The algorithm uses the largest (maximum) DNA percentage available from one of the children. In the scenario being examined here, this results in the minimum total coverage (37,5%) within the range so that it is effectively a lower bound. As noted in the text, the most frequent combination within the range (the average of the two ends) is what Paul Woodbury's formula uses.