160-2012

The Traveling Salesman Traverses the Genome:

Using SAS® Optimization in JMP® Genomics to build Genetic Maps

Kelci Miclaus, Rob Pratt, Matthew Galati

SAS Institute, Cary, NC

ABSTRACT

Agronomic research employs genomics to develop breeding strategies that improve the disease resistance, health, and yield of agricultural crops. Plant genome complexity necessitates understanding how DNA markers across the genome are related by building genetic maps that group and order markers based on their inheritance. This biological challenge is an ideal application of optimization routines common in operations research. Optimal marker order can be formulated as the famous traveling salesman problem, where you follow a path through the genome that returns the shortest genetic distance map. Minimum spanning trees can group markers using an undirected graph. Using algorithms from the OPTMODEL Procedure in SAS/OR® implemented in a visual interface with JMP Genomics, biologists can quickly and optimally build genetic maps.

INTRODUCTION

Genomic research is one of the fastest growing fields in science and technology. The first few human genomes were sequenced in 2001 at a cost of three billion dollars. This work formed the initial basis of the human reference genome, providing information about locations of variation along your chromosomes that lead to the diversity that humans exhibit. This year, there is the promise of a sequencing tool that can produce a fully sequenced human genome at an operating cost of $1,000 in just a day. Next-generation sequencing costs and technology are advancing at a rate that easily outpaces Moore's Law, greatly aiding the effort to understand the genetic underpinnings to human health (and disease).

Nutrition is a key component to human health, and it is no longer just a trite saying that "you are what you eat." Indeed, research has demonstrated that genomic material from plants you consume can drive the cellular processes in your body. Despite the explosion of research on the human genome, plant genomic research has a different set of difficult issues. Plant genomes are often more complex than the "simple" human genome and very few plants have been fully sequenced to provide reference genomes. In this case, an estimated genetic map is necessary in order to determine the locations (markers) throughout the plant genome that drive desirable traits such as crop yield, health benefits, disease resistance, and environmental stability. Marker-assisted breeding strategies involve targeted breeding of new plant varieties to increase desirable traits through genetic inheritance. Without such research, the effort to feed the growing world population will become increasingly difficult as agricultural land resources decrease.

A genetic map is an ordered set of DNA markers derived from their inheritance patterns in an experimental (inbred or controlled) population. From the molecular data collected in the experiment, computational algorithms must be employed to group, order, and space markers into an optimal "map" based on their how often markers are inherited together (genetic distance). As such, genetic map construction becomes analogous to solving the famous NP-hard combinatorial Traveling Salesman Problem (TSP). This paper discusses the biological and technical aspects of creating a genetic map to facilitate critical research. The problem poses a unique challenge because it relies on integration of expert biological knowledge with advanced optimization algorithms. The paper also highlights new tools available in JMP Genomics software to construct genetic linkage maps through a graphical user interface from which biologists can access powerful optimization routines implemented with SAS/OR procedures such as PROC OPTMODEL.

GENETIC MAP CONSTRUCTION

Construction of a genetic map is conceptually simple. When you do not know the physical distance between DNA markers across the genome, genetic distance is used as a surrogate and is inferred from the frequency at which the same markers are inherited together. If the (genetic) distance between two markers is very large, it is likely that the two markers are from separate chromosomes. When researchers first started forming genetic maps (nearly a hundred years ago), no more than a handful of markers were available, and it was not computationally intensive to determine the optimal marker order with exhaustive searches. An exhaustive search would require checking *n!* possible maps (sequences), where *n* is the number of markers. Advances in technology have changed such problems from 10s of markers to 1000s of markers, and an exhaustive search is now infeasible. Genetic maps are essentially built in two steps: assigning markers into meaningful groups (analogous to chromosomes), and ordering the markers within groups to minimize the overall genetic map distance. Solvers available in the OPTMODEL procedure can provide solutions to both steps with speed, efficiency, and flexibility that is unparalleled by currently available software packages.

BIOLOGICAL BACKGROUND

The focus of this paper is on the application of optimization techniques to solve a difficult biological problem. This section provides a brief background of the relevant molecular biology necessary to understand the problem. The following are definitions of key terms:

- *DNA:* The genetic code (made up of nucleotide bases A, T, C, G) that determines your unique physical characteristics. You inherit two copies of this code (one from your mother, one from your father).
- *Chromosome:* An organized structure of coiled DNA, the number of which vary among organisms (humans have 23).
- *Chromatid:* One half of a chromosome. It is a single copy of the genetic code (DNA) that is passed on to offspring from each parent.
- *SNP (Single Nucleotide Polymorphism):* A location in your code (DNA) that exhibits variation, e.g. one person may have an A base, another has a C. This is the most common type of DNA marker that is analyzed for association with diseases. These markers form the nodes of the genetic map.
- *Physical Distance:* Measured in number of bases, this is the actual physical distance between markers on the same chromosome
- *Cross-Over/Recombination:* The chromatid you inherit from a parent is never an exact copy. When the chromatid forms, the two copies of the parental DNA code can break and cross-over and recombine to make a new set of genetic code.
- *Genetic Distance:* Measured in a scale called Morgans or CentiMorgans, genetic distance is a function of the number of cross-over events (recombination frequency) that occurred between two markers in an experiment. Genetic distance can approximate physical distance since if two markers are physically close, the probability of a cross-over is very low.
- *Linkage:* If there are very few cross-overs between markers (low recombination) then the markers are said to be "linked" and marker variations tend to be inherited together. Markers in linkage are genetically close together (low genetic distance).
- *Framework Map:* A known grouping and/or ordering of markers based on previous biological knowledge or experiments that, when available, is used to define constraints when building a genetic map.

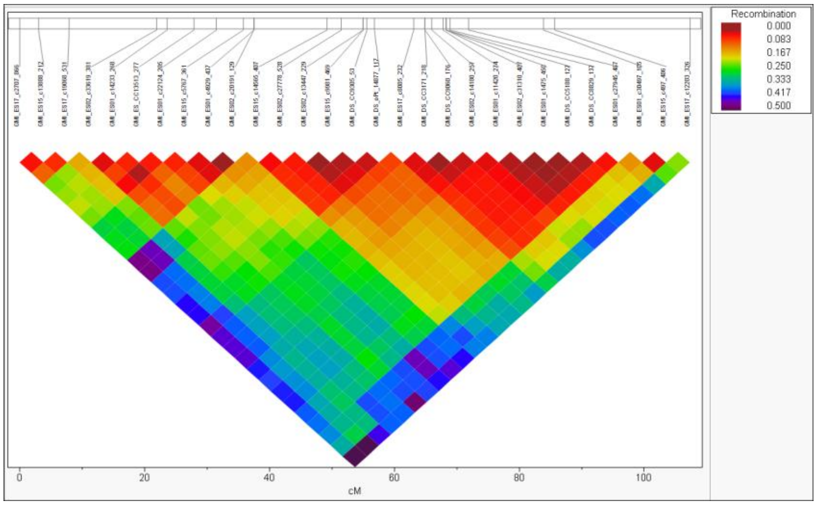

Figure 1: JMP Genomics plot of a genetic map for one linkage group coupled with the heat map of recombination distances from which it was derived.

In experimental populations, e.g. inbred plant lines, you know what markers are inherited from each parent and you can compute all pairwise marker recombination frequencies as a distance matrix. From these frequencies, you can infer which markers are linked. This distance matrix forms the data used to construct a genetic linkage map. Figure 1 is an example of a genetic distance map for a single group and was drawn using one of the new processes for linkage mapping in JMP Genomics 6.0. The horizontal line along the top represents the computed genetic distance for optimally ordered markers. The triangle color plot represents the corresponding recombination frequencies among the markers.

APPLICATION OF OPTIMIZATION CONCEPTS

The first step of genetic map construction is grouping the markers based on patterns of their recombination frequency into linkage groups. Typically, if markers have a nominally low recombination fraction (less than 0.25) then the markers are said to be linked. Standard hierarchical clustering techniques can be used to cluster markers, but often researchers have prior knowledge of markers that have been either anchored to a known chromosome or commonly grouped together from other experiments. Common clustering methods are not flexible enough to handle constraints such as framework markers belonging to known groups. In this case, a constrained minimum spanning forest can be used to group markers. Once markers have been grouped, a solution to the TSP has become an industry standard used to determine the order of the markers that minimizes the overall genetic distance. When a biologist knows that certain markers belong in a particular order (a framework map), the problem can be solved as a side-constrained TSP. Figure 2 is a view of a completed genetic linkage map, drawn using the Linkage Map Viewer process in JMP Genomics 6.0. Italicized markers represent framework markers with their colors corresponding to previously known orders (spectrally ranging from low to high as purple to orange).

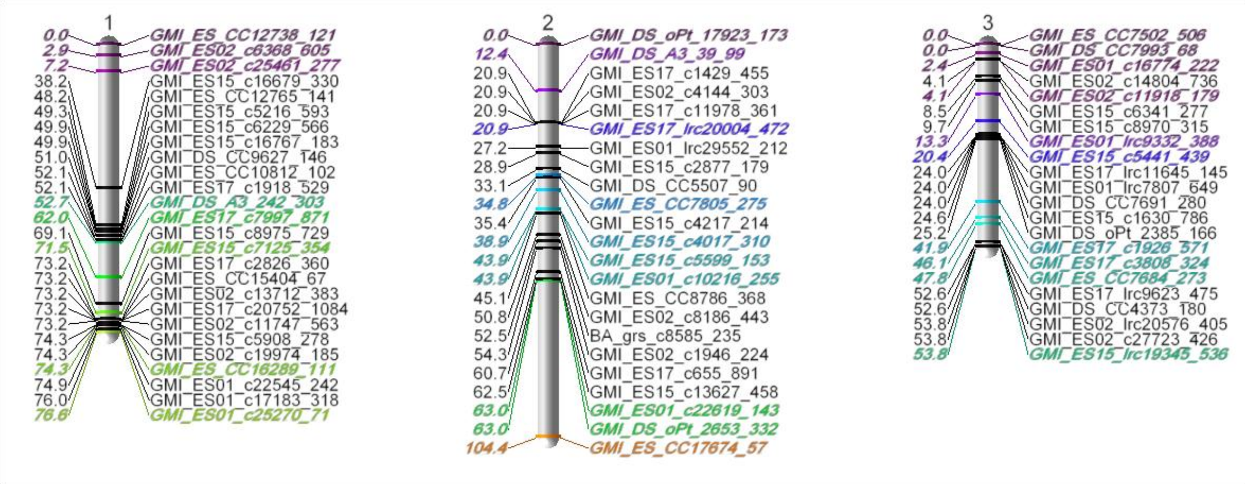

Figure 2: A graphical depiction of a genetic map generated using linkage mapping processes in JMP Genomics.

Marker Grouping

Most available software tools employ single linkage hierarchical clustering, which defines linkage groups based on the minimum recombination frequency (distance) between groups of markers. A more recently available tool called MSTmap (Wu et al. 2008) employs minimum spanning trees (MST) for genetic map construction. Single linkage clustering can be shown to be equivalent to the solution of a minimum spanning forest. You can formulate the linkage group problem as an undirected graph, where markers are nodes and edges are weighted by the recombination distance between the nodes. By terminating Kruskal's minimum spanning tree algorithm (Kruskal 1956) when the cost to connect the edge surpasses a threshold, you obtain a minimum spanning forest, and the connected markers at that point form your linkage groups. Using the OPTMODEL procedure in SAS/OR, you can generalize this solution to accommodate known linkage group membership among certain markers to disallow specific marker paths or to require certain connected nodes. This novel approach has been implemented in the upcoming release of JMP Genomics (Version 6.0).

Marker Order Optimization

Formulating the genetic map marker ordering problem as a TSP is a well-known approach. As the number of available sequenced markers started growing, optimal solutions obtained through exact methods such as the branch-and-bound strategy were replaced with local search methods for speed and computational ease. Common software tools for linkage mapping employ simulated annealing (Jansen et al. 2001) or genetic algorithms such as guided evolutionary strategy (Mester et al. 2003) and ant colonization (Iwata and Ninomiya, 2006). These methods provide solutions quickly, however they cannot guarantee optimality. In fact, they cannot even give a bound on the quality of the solution. As these dense genetic maps become increasingly available and more integral to research, an optimal solution is extremely desirable if it can be carried out in a reasonable amount of time. Many of the current software providers for this biological problem do not have the optimization expertise or the resources to efficiently implement optimal genetic map solutions. The Mixed Integer Linear Programming (MILP) solver integrated into PROC OPTMODEL can provide optimal solutions for the marker ordering problem with efficiency and flexibility that is unparalleled in existing software solutions.

GENETIC MAPS WITH JMP GENOMICS AND SAS/OR®

JMP Genomics is a bundled software solution of JMP and SAS for genomic data analysis. User-driven dialogs are created in the point-and-click JMP environment to set up custom analyses, and extensive SAS macros are employed under the hood to perform the analysis using SAS software. The results of each analysis are launched back to JMP and typically include both advanced statistical output and interactive dynamic graphics. Three new processes in JMP Genomics software were developed for genetic linkage map construction: Recombination and Linkage Groups, Linkage Map Order, and Linkage Map Viewer. The first two of these perform linkage grouping and marker order optimization and tap into the advanced solvers available within the OPTMODEL procedure in SAS/OR. Under this software paradigm, you can implement complex optimization routines in an integrated interface that is tailored for biological applications.

The remaining sections of the paper highlight JMP Genomics capabilities using genetic marker data from an oat research experiment. The data consist of 467 markers for 90 plant lines derived from a cross of two oat varieties. The data were kindly provided by the Collaborative Oat Research Enterprise (CORE), a USDA-led international effort to increase the health benefits in oat.

LINKAGE GROUPING

The Recombination and Linkage Groups process computes the recombination distance matrix and performs subsequent clustering/grouping from the input marker data and biological specifications for an experimental population. Numerous parameters can be entered in the JMP Genomics user interface, which has been developed specifically for use by biologists. These include parameters such as the minimum distance with which to define groups, the algorithm to use for grouping (hierarchical clustering options and minimum spanning tree solution), and existing biological constraints (framework markers). Once the biologist runs the process and the SAS program has completed, the recombination matrix and group membership are displayed using interactive graphics to elucidate patterns of the data.

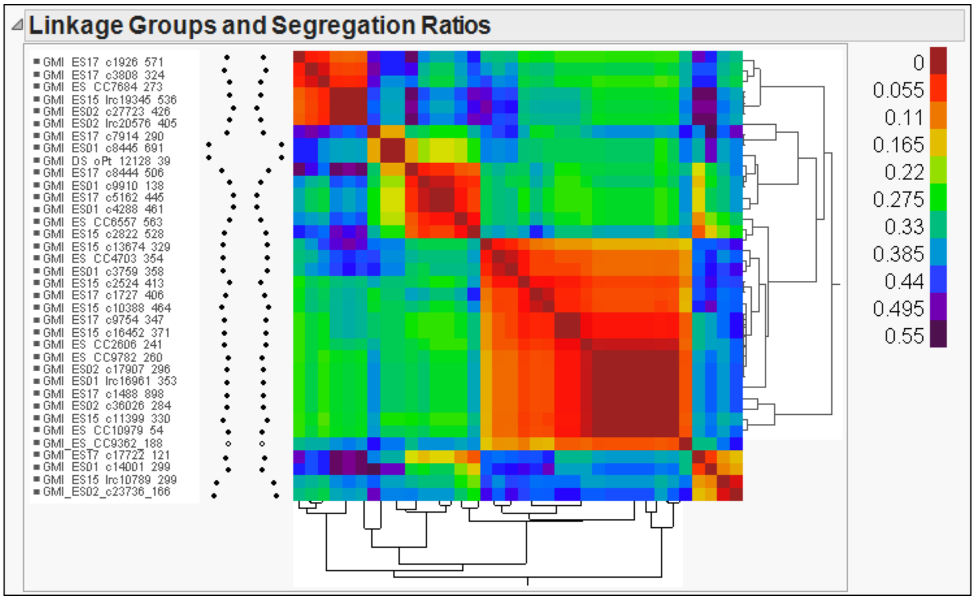

Figure 3: Graphical results in JMP Genomics of a constructed linkage group based on the MST SAS/OR solution

Figure 3 is an example of a derived linkage group based on a minimum recombination threshold of 0.25. The use of the hierarchical clustering platform in JMP shows that there may be further structure in this linkage group and that it may be desirable to use a more stringent distance threshold for grouping. This is one of the 22 linkage groups that were formed by the minimum spanning forest solution with the CORE oat data (oat has 21 chromosomes).

Minimum Spanning Forest Details

The minimum spanning forest solution is most useful when a set of the markers belong to known groups. This is common biological knowledge obtained either from prior experiments that have anchored markers to a known chromosome or from previous genetic maps for a species built from several experiments (consensus maps). The oat population analyzed from the CORE project is part of several populations for which a consensus map is built; therefore several of the markers are known to belong to framework groups. The solution in PROC OPTMODEL builds in constraints based on the framework when considering an edge between nodes.

The following OPTMODEL statements implement a minimum spanning forest algorithm. No mathematical optimization solver is called; instead, the programming statements provided by OPTMODEL can be altered to iteratively include the currently minimum cost edge that connects two different components. The algorithm terminates when the number of components decreases to the user-defined threshold `&lgthresh` or when all remaining admissible edges have cost at least as large as the user-defined threshold `&rfthresh`, whichever comes first.

```
/* Sort step to decrease computation time in search for minimum spanning forest */
proc sort data=&indata;
 by &cost;
run;

proc optmodel;
 set <str>     NODES;
 set <str,str> EDGES;
 num cost {EDGES};
 set <str,str> TREE_EDGES init {};

 /* read edges from data set */
 read data &indata into EDGES=[&marki &markj] cost=&cost;
 NODES = union {<i,j> in EDGES} {i, j};
 num component_id {NODES};

 /* read component_id in from framework data */
 read data &sdata into [&marki] component_id=_skeleton_;
 num num_components init max {i in NODES} component_id[i];
 set <str> COMPONENT {1..num_components} init {};

 num c;
 for {i in NODES} do;
    c = component_id[i];
    COMPONENT[c] = COMPONENT[c] union {i};
 end;

 /* if i and j in different components, merge components and add tree edge */
 num ci; num cj;
 set SKELETON = {i in NODES: component_id[i] <= &lgthresh};
 set TEMP_EDGES init {<i,j> in EDGES: i not in SKELETON or j not in SKELETON};
 num min_cost;
    do until (card(setof {n in NODES} component_id[n]) <= &lgthresh or
             (min_cost >= &rfthresh));
    for {<i,j> in TEMP_EDGES} do;
       min_cost = cost[i,j];
       ci = component_id[i];
       cj = component_id[j];
       if (ci ne cj) and (ci > &lgthresh or cj >&lgthresh) then do;
          /* update the smaller numbered component to add to framework groups*/
          if ci > cj then do;
             for {k in COMPONENT[ci]} component_id[k] = cj;
             COMPONENT[cj] = COMPONENT[cj] union COMPONENT[ci];
             COMPONENT[ci] = {};
          end;
          else do;
             for {k in COMPONENT[cj]} component_id[k] = ci;
             COMPONENT[ci] = COMPONENT[ci] union COMPONENT[cj];
             COMPONENT[cj] = {};
          end;
          /* add edge to tree */
          TREE_EDGES = TREE_EDGES union {<i,j>};
       end;
       TEMP_EDGES = TEMP_EDGES diff {<i,j>};
       leave;
    end;
 end;
 create data outgroups from [&marki] = NODES LGroupnum = component_id;
quit;
```

The data read into PROC OPTMODEL contains the pairwise recombination matrix (`&indata`) and defines the edges and nodes of the graph. The framework/skeleton groups (`&sdata`) are read to initialize the `COMPONENT` set. The set of feasible edges honoring framework group membership is defined by the `TEMP_EDGES` set and used to conditionally evaluate node connection (joining markers to groups). When the algorithm completes, the `component_id` vector defines linkage group membership for the nodes (markers).

MARKER ORDER OPTIMIZATION

The Linkage Map Order process takes as input the recombination distance matrix and group membership generated from the Linkage Groups process. Very few parameters need to be surfaced to the interface in order to execute a macro to solve the TSP marker ordering problem with PROC OPTMODEL. The dialog gives a biologist options to specify an existing framework map (if some markers are known to be in a certain order), a threshold constraint for edges to consider, and early stopping rules to decrease computational time by returning a near-optimal solution. Figure 4 shows the results in JMP Genomics for the Linkage Map Order process with the CORE oat data. The genetic distance calculated by the process (based on the optimal marker order solution) is plotted on the Y-axis for each of the linkage groups. Color is used to flag markers that were from a framework map (known groups and order). Interactive linkage group plots (as shown in Figure 1) are also produced as part of the results of this process.

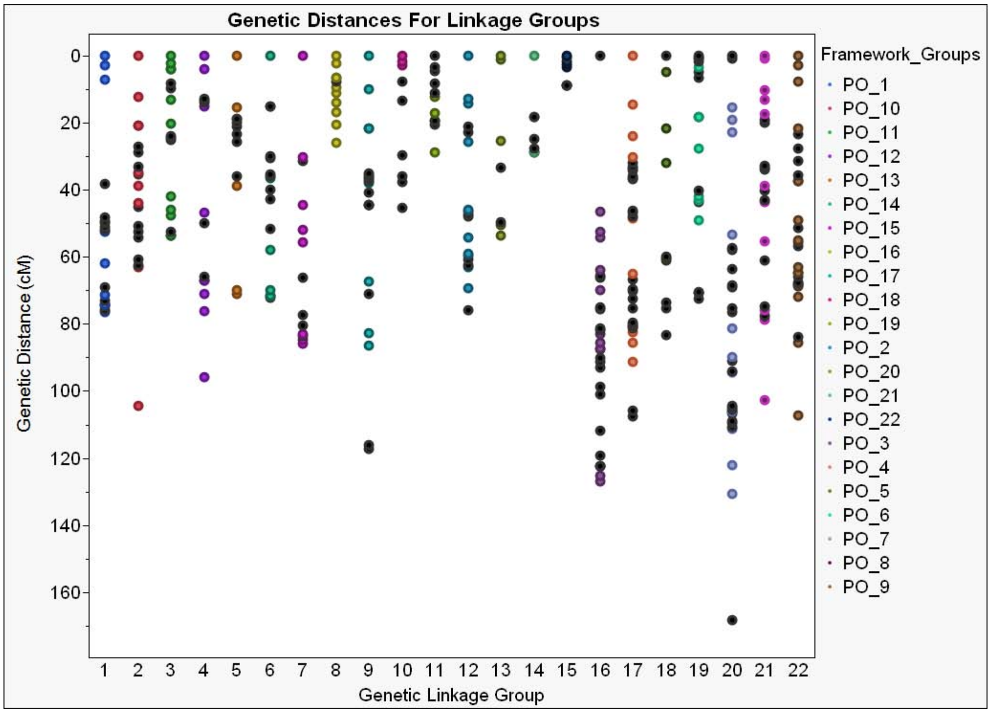

Figure 4: Graphical output generated by JMP Genomics to display the marker order solution and genetic maps provide clear and intuitive results for biological researchers.

Computational Details for the CORE Oat Data

For the 467 markers (placed across 22 linkage groups) in the CORE oat data, the process (data manipulation, solving in PROC OPTMODEL, and report generation) completed in under a minute and a half on a standard desktop computer. The median time spent by PROC OPTMODEL was 0.2 seconds for the 22 linkage groups. Table 1 displays the number of markers in the 22 linkage groups and the computational time spent for the CORE oat data. Note that an anomalous linkage group (Group 19) required considerably more computation time (62.4 seconds) in this example. For these data, constraints to honor known marker framework orders required more computational time/resources in comparison with an unconstrained TSP solution. The total processing time for the 22 linkage groups when using the side-constrained TSP solution was 71.55 compared to 3.87 seconds without constraints for the framework order map.

Computation Times with CORE Data

Linkage Group	Number of Nodes	Time (seconds)
1	26	0.37
2	23	0.21
3	22	0.26
4	15	0.08
5	13	0.04
6	18	0.17
7	15	0.07
8	14	0.05
9	37	3.04
10	10	0.04
11	18	0.19
12	22	0.29
13	7	0.01
14	9	0.03
15	8	0.03
16	31	0.75
17	37	0.82
18	10	0.02
19	29	62.4
20	43	2.1
21	28	0.31
22	29	0.27

Table 1: Computation times for the PROC OPTMODEL marker order solutions with the CORE oat data

For these data, an average 2.7% decrease in the resulting genetic map distance was observed across comparable ordered maps generated with guided evolutionary strategy, which is the solution implemented in another software program that is often considered the gold-standard for marker order solutions. It should be noted that these data are very well chosen, high-quality, genetic markers. In experience with several other data sets, up to a 30% decrease in the genetic map distance has been observed. As shown in the table above, the OPTMODEL procedure solution is both computationally efficient and can guarantee an optimal solution where other software tools cannot.

OPTMODEL Procedure Technical Details

The following OPTMODEL statements, based on the TSP example in the MILP solver documentation, implement a cutting-plane algorithm that dynamically generates violated subtour elimination constraints. In each stage, the MILP solver is called with the current set of constraints. Once the support graph of the resulting solution is connected, the algorithm terminates with an optimal TSP tour.

```
%macro findConnectedComponents;
/* find connected components of support graph */
…
%mend findConnectedComponents;

%macro rowGenerationLoop;
       /* generate constraints dynamically as needed */
      do until (numcomp = 1);
       iter = iter + 1;
         /* solve with current set of constraints */
         solve with milp %if %upcase(&UseStoppingRules)  = YES
                         %then / &relobjgap1 &maxtime;;
       %postSolve;
         %findConnectedComponents;
         if (numcomp = 1) then do;
         iter = iter + 1;
             /* if no violations, then solve to optimality */
             put "Solving to optimality...";
             solve with milp / primalin %if %upcase(&UseStoppingRules) = YES
                                         %then    &relobjgap2 &maxtime;;
         %postSolve;
             %findConnectedComponents;
         end;
      end;
%mend rowGenerationLoop;

proc optmodel presolver=none;
 set <str> VERTICES;
 set <str,str> EDGES = {i in VERTICES, j in VERTICES: i > j};
 num numsubtour init 0;
 set <str> SUBTOUR {1..numsubtour};
 read data &indata into VERTICES=[&IdVar];
 VERTICES = VERTICES union {'DUMMY'};
 num c {VERTICES, VERTICES} init 0;
 read data &indata into [&IdVar]
    {j in VERTICES diff {'DUMMY'}}<c[&IdVar,j]=col(j)>;

 /* x[i,j] = 1 if edge (i,j) is traversed, 0 otherwise */
 var x {EDGES} binary;
 /* minimize the total cost */
 min obj = sum {<i,j> in EDGES} c[i,j] * x[i,j];
 /* each vertex has exactly one in-edge and one out-edge */
 con two_match {i in VERTICES}:
    sum {j in VERTICES: i > j} x[i,j]
    + sum {j in VERTICES: i < j} x[j,i] = 2;
 /* no subtours (these constraints are generated dynamically) */
 con subtour_elim {s in 1..numsubtour}:
    sum {<i,j> in EDGES: (i in SUBTOUR[s] and j not in SUBTOUR[s])
    or (i not in SUBTOUR[s] and j in SUBTOUR[s])} x[i,j] >= 2;
  num lower_bound init 0;
  con lower_bound_def:
      obj >= lower_bound;

 for {<i,j> in EDGES: c[i,j] > &threshold} fix x[i,j] = 0;

 /* this starts the algorithm to find violated subtours */
 set <str,str> EDGES1;
```

```
  set <str> INITVERTICES = setof{<i,j> in EDGES1} i;
  set <str> VERTICES1;
  set <str> NEIGHBORS;
  set <str,str> CLOSURE;
  num component {INITVERTICES};
  num numcomp;
  num iter init 0;
  set ITERS = 1..iter;

  %rowGenerationLoop;

  set <str,str> SUPPORT init {<i,j> in EDGES: round(x[i,j].sol) = 1};
  create data solData from [i j]=SUPPORT c;

  /* traverse arcs to determine Order */
  num Order {VERTICES};
  str curr_vert init 'DUMMY';
  Order['DUMMY'] = 0;
  for {curr_order in 1..card(VERTICES)-1} do;
     for {<i,j> in SUPPORT: curr_vert in {i,j}} do;
        for {other_end in {i,j} diff {curr_vert}} do;
           Order[other_end] = curr_order;
           curr_vert = other_end;
        end;
        SUPPORT = SUPPORT diff {<i,j>};
        leave;
     end;
  end;
  create data solData2 from [&IdVar]=(VERTICES diff {'DUMMY'}) Order;
quit;
```

Completed Genetic Linkage Map

The final new process for genetic linkage map construction in JMP Genomics is the Linkage Map Viewer. Custom graphical scripting available in JMP creates a sophisticated visual representation of the genetic map shown in Figure 5. The process gives you options to annotate the markers based on labels, color, and fonts. In this application area, good graphical aids are critical for biologists to understand and convey the results derived by advanced optimization methods. This user paradigm plays to the strengths of integrating JMP (as an excellent, intuitive visual analysis platform) and SAS/OR (as an extremely powerful optimization environment) for the unique inter-disciplinary problem posed by genetic map construction.

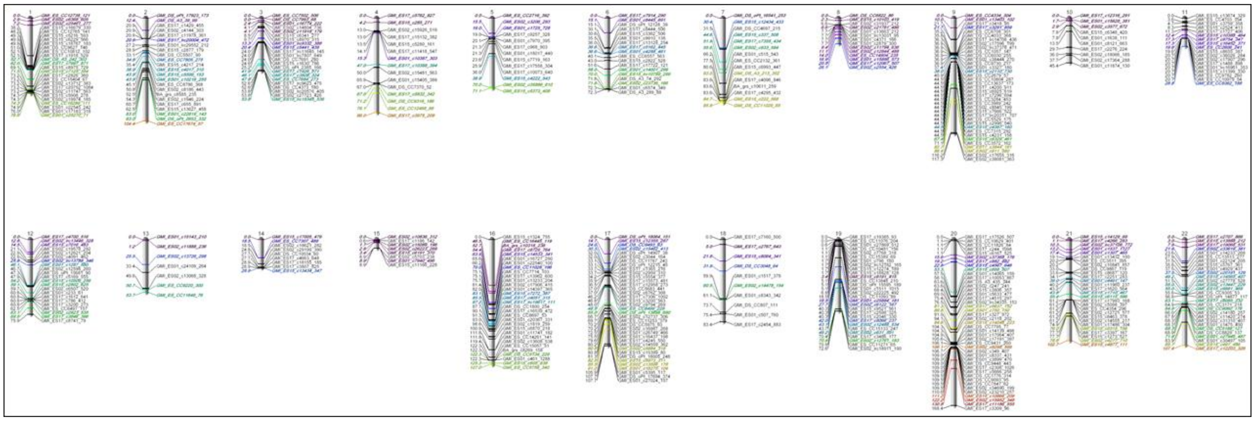

Figure 5: The genetic map constructed with JMP Genomics and SAS/OR for the CORE oat data.

THE FUTURE IS BRIGHT (AND FAST)

The next release of SAS/OR (Fall 2012) includes a specialized TSP solver, available both in PROC OPTMODEL and in the new OPTNET procedure. This TSP solver is even more sophisticated and can find optimal solutions much faster than the generic MILP solver. It can therefore be useful for solving larger instances or for running simulations that rely on solving TSPs multiple times. Larger marker problems on the order of 250-500 markers that took several hours to return an optimal solution with the current MILP solver in OPTMODEL can be completed in seconds with the specialized TSP solver. Furthermore, the OPTNET procedure offers programming ease of use for this application in comparison with the more general OPTMODEL programming environment. The OPTNET procedure can be written in less than 10 lines to formulate the TSP solution, while it took several lines of code in OPTMODEL (as shown in the OPTMODEL Procedure Technical Details)

The availability of such an efficient, optimal solver will expand the application of SAS/OR tools in this arena. A common issue that plagues map construction is marker data quality. With the upcoming tools, it will be possible to iteratively sample from the data and create hundreds of solutions to identify poor quality markers that disrupt the map solution or that are unstable. The use of SAS/OR tools can also be extended to building consensus genetic maps that combine and optimally unify genetic maps from multiple experiments.

CONCLUSION

The marriage of the optimization capabilities in SAS/OR tools with the visually intuitive JMP Genomics framework provides an excellent tool for biologists, geneticists, and bioinformaticians constructing genetic linkage maps. The research area presents unique problems that require integration of domain and technical expertise that most available software tools cannot provide. These new applications epitomize the value of combining robust analysis techniques (like those in SAS/OR) with domain-specific interface tools (like JMP Genomics) offered by SAS.

REFERENCES

Cheema, J, Dicks, J. 2009. “Computational approaches and software tools for genetic linkage map estimation in plants.” *Briefings in Bioinformatics.* **10(6)**: 595-608

Wu Y, Bhat PR, Close TJ, Lonardi S. 2008. “Efficient and Accurate Construction of Genetic Linkage Maps from the Minimum Spanning Tree of a Graph.” PLoS Genet 4(10): e1000212. doi:10.1371/journal.pgen.1000212

Jansen, J, De Jong, AG, van Ooijen, J.W. 2001. “Constructing dense genetic linkage maps.” *Theoretical and Applied Genetics*. **102**: 1113-1122.

Kruskal, JB. 1956. “On the shortest spanning subtree of a graph and the traveling salesman problem.” Proceedings of the American Mathematical Society, **7(1)**: 48–50.

Mester D, Ronin Y, Minkov D, Nevo E, Korol A. 2003. “Constructing large scale genetic maps using evolutionary strategy algorithm.” *Genetics*, **165:** 2269-2282.

Iwata H, Ninomiya S. 2006. “AntMap: constructing genetic linkage maps using an ant colony optimization algorithm.” *Breeding Science.* **56**:371-377

ACKNOWLEDGMENTS

The authors would like to thank Eric Jackson and other members of the CORE (Collaborative Oat Research Enterprise) for providing the data highlighted in this paper as well as domain expertise and advice with genetic map construction.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Kelci Miclaus
Enterprise: SAS Institute Inc.
Address: 100 SAS Campus Dr.
City, State ZIP: Cary, NC 27513
Work Phone: +1 919-413-8004
E-mail: Kelci.Miclaus@sas.com