COMPUTERIZING AND STREAMLINING OSA

For the purpose of OSA (sensu Colbert and Rowe 2008), semaphoronts can be considered bit arrays (i.e., bitstrings). OSA establishes the most-parsimonious ontogenetic sequences based on semaphoront scores for a particular series of irreversible events. The set of considered events (either binary and/or multistate) defines a hypercube, on which the observed and hypothesized semaphoronts and sequence segments can be graphically positioned and measured.

The hypercube describes all possible paths between all possible semaphoronts (i.e., edges between vertices) and is a heuristic for establishing standardized distances between populations of semaphoronts. Its potential to formalize a topological description of the OSA map is a major avenue for future statistical comparison between diverse populations (e.g., different taxa, different genders, different experimental manipulations, etc.) that potentially occupy distinct regions of the hypercube. To understate it, the ability to measure sequence differences between samples would be an incredible asset for comparative and developmental biologists.

The following presents some ideas for the revised OSA technique. This first includes a ‘terminology’ section that defines some basic methodology and concepts. The new terms that I have coined may not be necessary, but I found them useful in describing the analytic method. The method itself is presented in the context of a hypothetical example.

Please note that work on the revised OSA is in progress. Areas requiring further investigation include:

1) The practicalities of developing a GUI and whether this involves development through the Mesquite environment. My preliminary exposure to Mesquite has left me underwhelmed.
2) The programming behind graphical (and potentially interactive) representation of the semaphoront’s inherent coordinate system (i.e., the OSA map).
3) The descriptive statistics that will be used to define the *n*-dimensional space of the OSA sequences (e.g., an *n*-dimensional centroid? perhaps some of the Blob stats might apply?). Additional descriptive statistics might include defining the *n*-dimensional space (which is similar to a convex hull) for particular events, for select groups of events, or even for missing data.
4) The matrix (see discussion below).
5) The feasibility of Mesquite for development of the revised OSA algorithm.
6) The establishment of standardized protocols and abbreviations for names of semaphoronts and hypothetical semaphoronts (i.e., something that the program would generate, and that would be interpretable by people other than seek-geeks (i.e., sequence geeks).

Basic concepts

Semphoront: The expressed ontogenetic or developmental state of an individual at an instant in its lifespan (sensu Hennig 1966). In OSA, the semaphoront is characterized by a string of scored irreversible events. The events are either discrete, or have been discretized from a continuous distribution (a possibility that needs further investigation). The OSA semaphoront is a type of bitstring.

Semaphoronts are either observed (i.e., represented by one or more specimens) or are hypothetical (perhaps there should be a name for hypothetical semaphoronts - such as 'hyporonts'). Hypothetical semaphoronts comprise any possible bitstring permutation – including permutations that would not be considered parsimonious by OSA.

Semaphoronts are here designated by the abbreviation ‘Sm’.

Example: SmA: 1110001. In this example, the binary string scores for seven ontogenetic events with ‘0s’ being the untransformed state, and ‘1s’ representing the transformed state. This string can be codified to describe multistate characters.

Maturity score: The sum of all event scorings in a particular semaphoront.

Example 1: SmA: 1110001. Maturity score = 4
SmB: 1010001. Maturity score = 3
SmC: 1110010. Maturity score = 4
SmD: 1210010. Maturity score = 5

1) Maturity score can be graphically used as a major axis with which to sort semaphoronts (I haven't seen any hypercube illustrations that use this convention).

Example 2:

MATURITY SCORE	Semaphoronts (as above)
5…………….	SmD
4……............	SmA, SmC
3…………….	SmB

2) Maturity score sorts are used to determine the **maxoronts** and **minoronts** (see definitions and discussion below).

3) Maturity score serves as a filter for optimizing analyses (see discussion below).

Pairwise distance: the sum of absolute values of the difference of all comparable events between two semaphoronts. This follows the general formula:

$$d_1\,(\mathrm{Smp},\,\mathrm{Smq}) = \sum_{i=1}^{n} |Smp_i - Smq_i| \text{ , where:}$$

$$\mathrm{Smp} = (Smp_1,\, Smp_2,\, ...,\, Smp_n) \text{ and } \mathrm{Smq} = (Smq_1,\, Smq_2,\, ...,\, Smq_n)$$

Example 3:

SmA: 111000
SmB: 110001
Semaphoront distance: 2.

Delminoront: The pair of semaphoronts that have the shortest pairwise distance within a considered population of semaphoronts. More than one pair of semaphoronts can be delminoronts in a population (i.e., can have equally short pairwise distances).

Example 5: In the considered set of the following three semaphoronts,

SmA: 111001 (maturity score 4)
SmB: 110001 (maturity score 3)
and SmC: 200001, (maturity score 3)

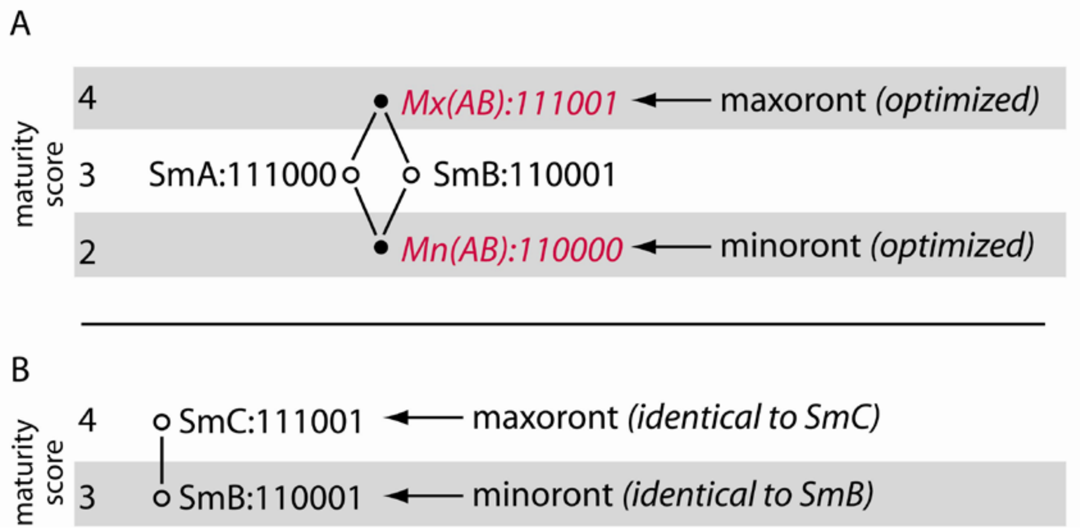
The pair comprising SmA & SmB are *delminoronts* because they show the shortest distance (1 step), while Sm C is 3 steps different from A, and is 2 steps from B.

Maxoront and Minoront: Every pairwise comparison of semaphoronts determines both a distance, a network of possible paths, including the identity of both maximum maturity-score semaphoront (maxoront) and minimum maturity-score semaphoront (minoront) that necessarily are involved in the sequence (see Figure 1). The maxoront and minoront can be identical (if the compared pair is identical).

The **maxoront** (here designated by '**Mx**') is defined relative to the optimization of the shortest paths in a pairwise comparison of semaphoronts. The maxoront is the one semaphoront that has the highest maturity score of all semaphoronts (observed or hypothesized) that occupy the shortest path or paths connecting a particular pair of semaphoronts (see Fig. 1).

The **minoront** (here designated by '**Mn**') is defined relative to the optimization of the shortest paths in a pairwise comparison of semaphoronts. The minoront is the one semaphoront that has the lowest maturity score of all semaphoronts (observed or hypothesized) that occupy the shortest path or paths connecting a particular pair of semaphoronts (see Figure 1).

Figure 1



.

Colbertoront: one who coins incredibly annoying names for arcane concepts.

A long these lines, do we need/want to use **hyporont** instead of 'hypothetical semaphoronts'?

The Matrix

Scoring the matrix

As before, the analysis starts with the scoring of an event-by-individual matrix for irreversible events.

We need to decide what sort of matrix would be best: and how much effort should be devoted to data entry. One extreme might be as simple as offering the capability of entering or uploading a tab-delimited format (e.g., Excel, or perhaps Nexus so that the phylogeny crowd would get their stroke).

Alternatively, we could develop a custom interface capable of relating scored specimens to ancillary information (metadata). These additional data could include optional data fields for: age (if known); gender; linear or volumetric measurements; whether experimentally manipulated (genetically or surgically); geographic information; pedigree; or any variable that could correlated be with OSA results. The ability to quickly sort the semaphoronts based on stored metadata would be a powerful tool.

Finally, the OSA program should be designed to accept longitudinal data. OSA analysis of these data can predict sequences that were not observed in the original data, and also establish likelihood estimates for sequences and events (e.g., see my analysis of Garn's data in Colbert and Rowe 2008). And, of course, it would provide the OSA map as a graphic and heuristic tool.

A REVISED METHODOLOGY

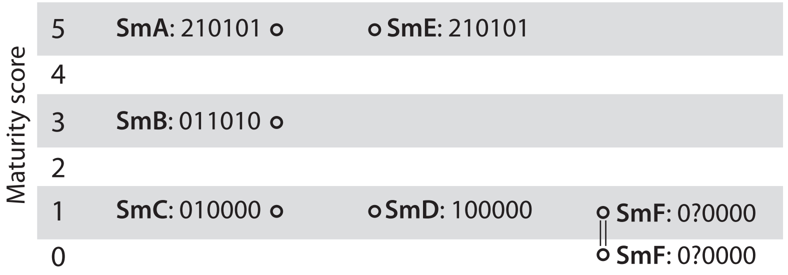
The revised OSA no longer requires PAUP or MacClade (which appears to be veering towards extinction). We will have to investigate the feasibility of OSA as a Mesquite module. My current interactions with Mesquite have been underwhelming.

The following example illustrates the basic steps involved in determination of the most-parsimonious maps and sequences.

REVISED OSA EXAMPLE ANALYSIS

Consider a very small OSA data set that includes six sampled individuals: **SmA**: 210101; **SmB**: 011010; **SmC**: 010000; **SmD**: 100000; **SmE**: 210101; **SmF**: 0?0000. Sample **SmF** was not scored for one event.

1) Sort individuals by **maturity score.** All observed data are indicated by open circles.



Open circles connected by a double-line indicate the possible range of maturity scores for **SmF**.

2) Calculate **pairwise distances**, identifying **delminoronts**. Missing bits of data (i.e., '?') are considered identical to the scored values to which they are compared.

SmA vs **SmB**: 6 steps
SmA vs **SmC**: 4 steps
SmA vs **SmD**: 4 steps
SmA vs **SmE**: **0 steps (delminoront)**
SmA vs **SmF**: 4 steps
SmB vs **SmC**: 2 steps
SmB vs **SmD**: 4 steps
SmB vs **SmE**: 6 steps
SmB vs **SmF**: 2 steps
SmC vs **SmD**: 2 steps
SmC vs **SmE**: 4 steps
SmC vs **SmF**: **0 steps (delminoront)**
SmE vs **SmF**: 4 steps

3) Synonymize semaphoronts for delminoronts having score of '0'.

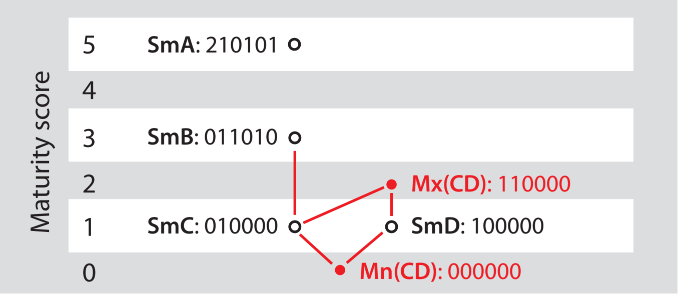
A) If the synonymized semaphoront has no missing data, it is apportioned a weight of '2.0' for statistical calculations.
(i.e., **SmA** = **SmE** synonymized to **SmA**; weight =2.0)

B) Synonymized semaphoronts having missing data are considered identical, but are retained for further comparison to newly optimized hypothetical semaphoronts.
Weight is not apportioned awaiting further analysis.
(i.e., **SmC** synonymized with **SmF** (in part))

4) The next **delminoronts** are identified, and their **maxoronts** and **minoronts** are established. By convention analysis begins with the delminoronts having the lowest maturity score. Sequence segments are drawn between maxoronts and minoronts of the delminoronts. Hypothetical semaphoronts are indicated by closed circles.

SmA vs **SmB**: 6 steps
SmA vs **SmC**: 4 steps
SmA vs **SmD**: 4 steps
SmB vs **SmC**: **2 steps (delminoront)**
SmB vs **SmD**: 4 steps
SmC vs **SmD**: **2 steps (delminoront)**

In this case, the maxoront and minoront for **SmB** vs **SmC** are identical to **SmB** and **SmC**, respectively.



Optimized semaphoronts and sequence segments shown in red.

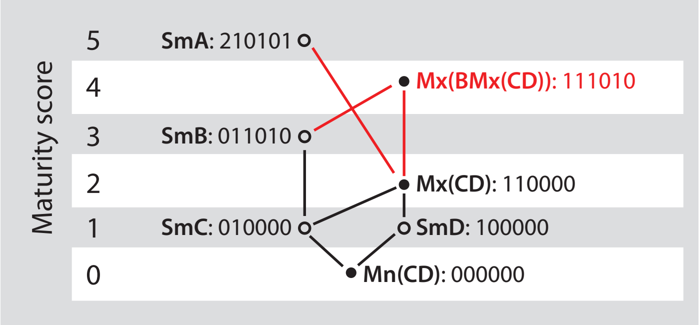
5) Pairwise distances are calculated for newly optimized semaphoronts (i.e., **Mx(CD)** and **Mn(CD)**). Pairwise comparisons need not be made with their 'parent' delminoronts (i.e., **SmC** and **SmD**). Similarly, the **SmF** optimization that is synonymous with **SmC** (i.e., missing event optimized as '1') does not require further comparison with the remaining semaphoronts.

SmA vs **Mx(CD)**: 3 steps
SmA vs **Mn(CD)**: 5 steps
SmB vs **Mx(CD)**: 3 steps
SmB vs **Mn(CD)**: 3 steps
SmF vs **Mn(CD)**: **0 steps**

SmF is now determined to be equivalent to **Mn(CD)**, in addition to **SmC** (see above)**.** **SmC** is apportioned a weight of 1.5, and **Mn(CD)** a weight of 0.5. Because all possible optimizations for the missing data of **SmF** have been considered, further pairwise comparisons no longer require inclusion of **SmF**.

6) **Maxoronts** and **minoronts** are established for the next **delminoronts.** Their connecting sequence segments are drawn.

SmA vs **SmB**: 6 steps
SmA vs **SmC**: 4 steps
SmA vs **SmD**: 4 steps
SmA vs **Mx(CD)**: **3 steps (delminoront)**
SmA vs **Mn(CD)**: 5 steps
SmB vs **SmD**: 4 steps
SmB vs **Mx(CD)**: **3 steps (delminoront)**
SmB vs **Mn(CD)**: **3 steps (delminoront)**



Newly optimized semaphoronts and sequence segments shown in red.

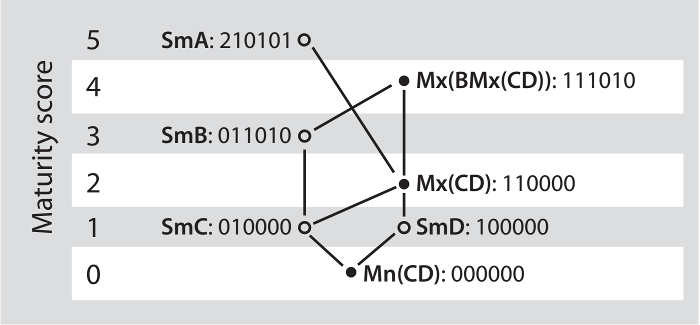
A) Analysis begins with delminoronts having the lowest maturity score: **SmB** vs **Mn(CD)**. The **SmB** vs **Mn(CD)** maxoront and minoronts are identical to **SmB** and **Mn(CD)**, respectively.

B) The next lowest maturity scores are **SmB** vs **Mx(CD)**. This comparison optimizes a new maxoront: **Mx(BMx(CD))**, but the minoront is identical to **SmC**.

C) Finally comparison is made with **SmA** vs **Mx(CD)**. The maxoront and minoront for **SmA** vs **Mx(CD)** are identical to **SmA** and **Mx(CD)**, respectively.

7) The newly optimized semaphoront is compared with all relevant semaphoronts, and the next **delminoronts** are identified. Their **maxoronts** and **minoronts** are established and sequence segments are drawn. If a pairwise comparison includes the same semaphoronts as an element in either pair, then that comparison is redundant and need not be performed (e.g., **SmB**, **SmC**, **SmD**, and **Mx(CD)** need not be compared to **Mx(BMx(CD))** because they are elements of both pairs).

SmA vs **SmB**: 6 steps
SmA vs **SmC**: **4 steps (delminoront)**
SmA vs **SmD**: **4 steps (delminoront)**
SmA vs **Mn(CD)**: 5 steps
SmA vs **Mx(BMx(CD))**: 5 steps
SmB vs **SmD**: **4 steps (delminoront)**



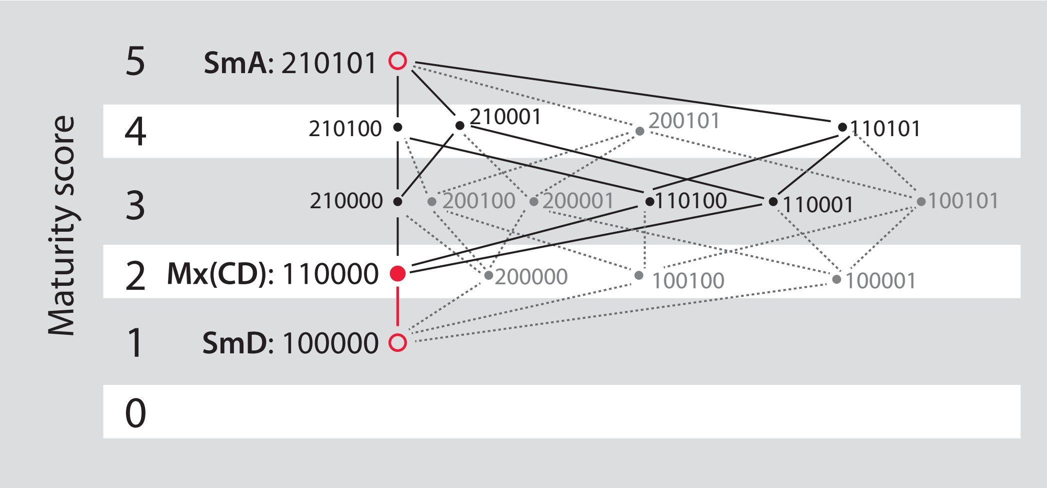
No new semaphoronts or sequence segments are defined by these comparisons.

A) The delminoront with the lowest maturity score is: **SmB** vs **SmD**. The maxoront and minoront for **SmB** vs **SmD** are identical to **Mx(BMx(CD))** and **Mn(CD)**, respectively.

B) The next lowest maturity scores are **SmA** vs **SmD**. The **SmA** vs **SmD** maxoront and minoront are the same as **SmA** and **SmD, respectively.** No new paths are drawn on the map, even though 24 paths have an equal length, and the difference establishes14 possible semaphoronts, one of which has been optimized as **Mx(CD)**. To minimize ad hoc pathways, the first segment of the sequence is considered resolved (i.e., event 2: 0 transforms into 1), leaving three unresolved events on this sequence (see Figure addendum).

C) Finally comparison is made with **SmA** vs **SmC**. The maxoront and minoront for **SmA** vs **SmC** are identical to **SmA** and **SmC**, respectively. Step '7' of this analysis did not result in any new semaphoronts or sequence segments.

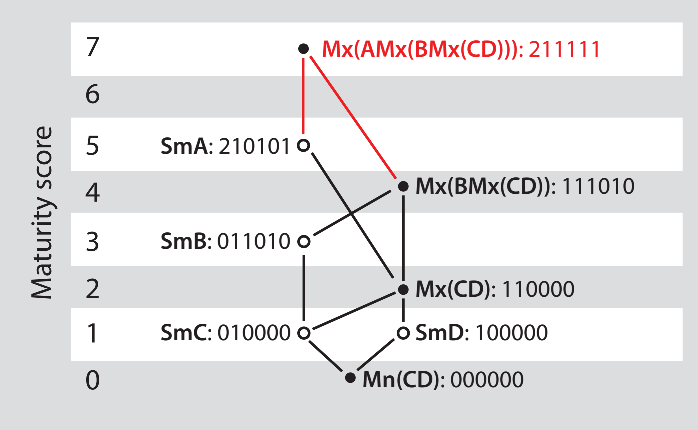
Figure addendum



Establishing the segment between **SmD** and **Mx(CD)** provisionally eliminates seven possible semaphoronts, and 18 possible sequences from consideration. Red lines and circles refer to observed semaphoronts **SmA** and **SmD**, and the optimized maxoront **Mx(CD)** . Solid black segments and circles represent unresolved sequences. Dashed gray line represents segments not warranted by observed data.

8) The next **delminoronts** are identified; and their **maxoronts** and **minoronts** are established and sequence segments are drawn.

SmA vs **SmB**: 6 steps
SmA vs **Mn(CD)**: 5 steps (delminoront)
SmA vs **Mx(BMx(CD))**: 5 steps (delminoront)

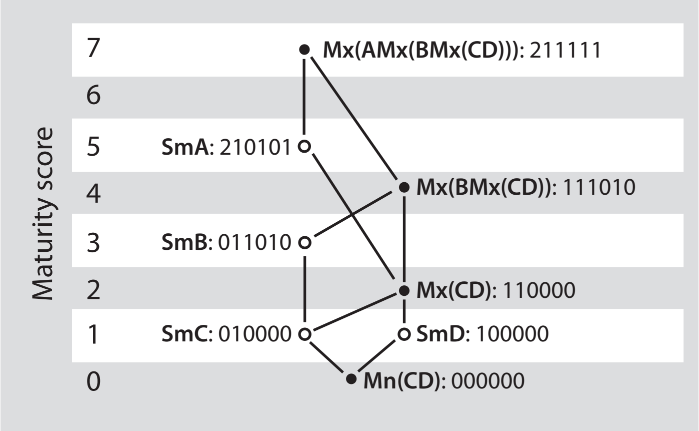


Newly optimized semaphoronts and sequence segments shown in red.

A) The delminoronts having the lowest maturity score are: **SmA** vs **Mn(CD)**. This comparison yields no new maxoronts or minoronts.

B) The next comparison is between **SmA** vs **Mx(BMx(CD))**. This comparison optimizes a new maxoront, **Mx(AMx(BMx(CD)))**, but the minoront is identical to **Mx(CD)**.

9) The last possible comparison is **SmA** vs **SmB**. This comparison yields no new semaphoronts. The newly optimized **Mx(AMx(BMx(CD)))**, has elements of all other possible comparisons and need not be further analyzed.
The basic map is done.



Now that the basic map has been recovered, the network of paths that comprise the sequences can be described, including sequence statistics. The map can also be redrafted to illustrate topological, demographic, and statistical properties, as well as to present an interactive interface for tracing characters and visualizing topologies and likelihoods.

10) Establish lists of all possible sequences.

A) Tally begins from the base of the tree (i.e., '000000')

By convention event strings are read from left to right (e.g., from events 1 to 6 here).
Arbitrarily optimize sequence building by looking for first changes starting from the left of the string of events.
Sequences are indicated by '**Sq**'.
In this case the first established sequence would be:

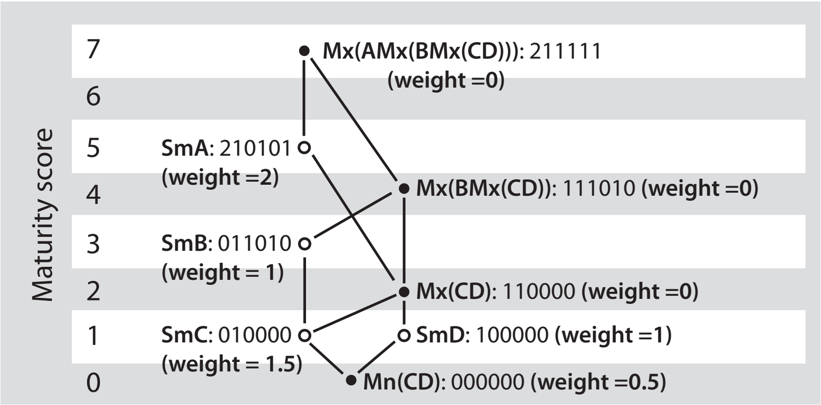
Sq1: 1:0-1→ 2:0-1→(1:1-2, 4:0-1, 6:0-1)→(3:0-1, 5:0-1)
then:
Sq2: 1:0-1→2:0-1→(3:0-1, 5:0-1)→(1:1-2, 4:0-1, 6:0-1)
then:
Sq3: 2:0-1→1:0-1 →(1:1-2, 4:0-1, 6:0-1)→(3:0-1, 5:0-1)
then:
Sq4: 2:0-1→1:0-1→ (3:0-1, 5:0-1)→(1:1-2, 4:0-1, 6:0-1)
then:
Sq5: 2:0-1→(3:0-1, 5:0-1)→1:0-1→ (1:1-2, 4:0-1, 6:0-1)

B) This can also be presented as a table (with tab-delimited output options).

In this case, the order of events arbitrarily conforms to the order seen in **Sq1**. Events that are not resolved in a particular sequence are assigned their average sequence position.

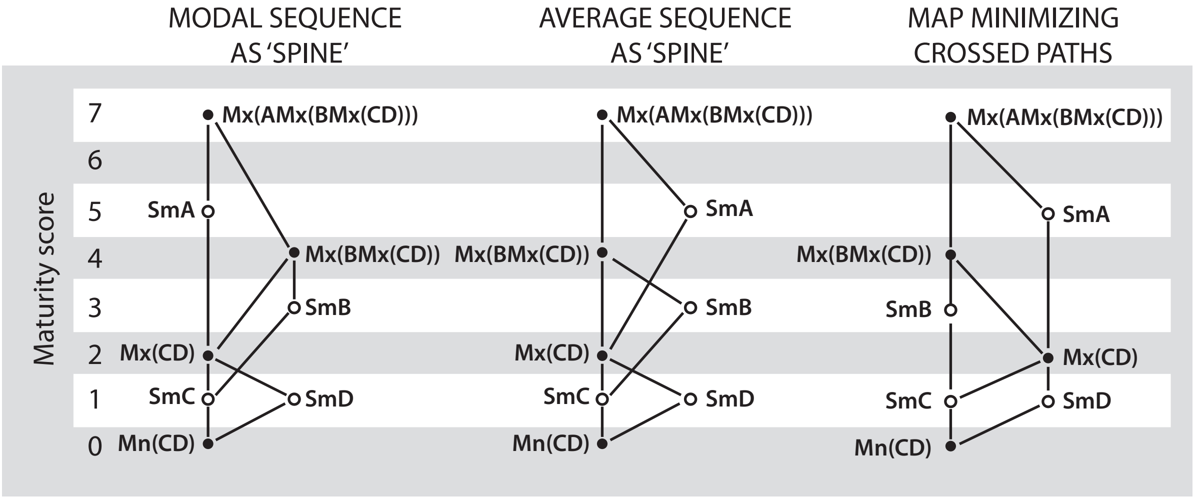
				events			
	1:0-1	**2:0-1**	**1:1-2**	**4:0-1**	**6:0-1**	**3:0-1**	**5:0-1**
sequences **Sq1**	1	2	4	4	4	6.5	6.5
Sq2	1	2	6	6	6	3.5	3.5
Sq3	2	1	4	4	4	6.5	6.5
Sq4	2	1	6	6	6	3.5	3.5
Sq5	4	1	6	6	6	2.5	2.5
average:	**2**	**1.4**	**5.2**	**5.2**	**5.2**	**4.5**	**4.5**
range:	**1-4**	**1-2**	**2-7**	**2-7**	**2-7**	**2-7**	**2-7**

11) The modal sequence is calculated by tallying all semaphoront weights on all sequences.



Modal sequence is **Sq3**, represented by 4 samples.
Next are: **Sq1** (3.5 samples); **Sq5** (3 samples); **Sq4** (2 samples); and **Sq2** (1.5 samples).

12) Sequence maps can be redrafted to conform 'spine' to average sequence, modal sequence, to a map that minimizes sequence path intersections, or to any conceivable distribution.



13) One of the main advantages of getting away from PAUP and MacClade is that we could better define the geometry of the of n-dimensional space upon which the sequences are predicted to occur by parsimony (i.e., estimating the shortest pathways, and minimizing the optimization of *ad hoc* pathways).