Supplemental Information

S1 Data Filtering

Filtering of the MDS annotation from Chen et al. "The Architecture of a Scrambled Genome Reveals Massive Levels of Genomic Rearrangement during Development" ***Cell*****, 158-5, (2014) 1187-1198.**

If the intervals $[a, a']$ and $[b, b']$ in a MIC DNA sequence correspond to the MDS annotations M and M' respectively, then M and M' are *intersecting* if $[a, a']$ intersects $[b, b']$ and *adjacent* if $b = a' + 1$ or $a = b' + 1$.

The dataset $\mathcal{D}$ is filtered from Chen et al. (2014) in the following way:

Step 1. Merge any two MDS annotations that are consecutive in a MAC contig and overlap or are adjacent in a MIC contig.

Step 2. Remove any two MDS annotations that are non-consecutive in a MAC contig and overlap or are adjacent in a MIC contig.

Step 3. Remove all duplicate MDSs, e.g., MDSs that correspond to MAC contigs that produced from alternative MIC processing.

Three processed files are included at `http://knot.math.usf.edu/data/scrambled_patterns/`.

1. processed_annotation_of_oxy_tri.gff contains all of the processed data $\mathcal{D}$.

2. list_of_DOWs_related_to_MICMAC_maps.txt contains all rearrangement maps, corresponding double occurrence words, and the reduced double occurrence words.

3. result_of_repeatreturn_pattern_application.txt contains the list of all 176 words that stabilize to non-empty words after performing the iterative repeat/return removal.

S2 Extended Table of Rearrangement Patterns

Reduced Rearrangement Patterns occurring in *O. trifallax*

Count	Reduced MIC pattern representative
854	$M_1\overline{M}_2$
307	M_2M_1
79	$M_1M_3M_2$
35	$M_1M_3M_2M_4$
34	$M_1M_3\overline{M}_2$
27	$M_1\overline{M}_2M_3$
24	$M_1M_3M_5M_7M_2M_4M_6M_8$
22	$M_1M_3M_5M_2M_4M_6$
21	$M_1\overline{M}_3M_2$
18	$M_2M_4M_1M_3$
17	$M_3\overline{M}_2M_1$
16	$M_1M_3M_5M_2M_4$
14	$\overline{M}_1M_3M_2$
14	$\overline{M}_2M_3M_1$
14	$M_1M_3\overline{M}_4\overline{M}_2$
14	$M_1M_3M_5M_7M_2M_4M_6$
13	$M_2\overline{M}_3M_1$
13	$M_3M_2\overline{M}_1$
11	$M_1M_3M_5M_7M_9M_2M_4M_6M_8$
11	$M_1M_3M_5M_7M_9M_{11}M_2M_4M_6M_8M_{10}M_{12}$
10	$M_1M_3M_5M_7M_9M_{11}M_{13}M_2M_4M_6M_8M_{10}M_{12}$
9	$M_3M_2M_1$
9	$M_1M_3M_5\overline{M}_6\overline{M}_4\overline{M}_2$
9	$M_1M_3M_5M_7M_9M_2M_4M_6M_8M_{10}$
7	$M_1M_3M_5\overline{M}_4\overline{M}_2$
7	$M_1M_3M_5M_7M_9M_{11}M_{13}M_{15}M_2M_4M_6M_8M_{10}M_{12}M_{14}M_{16}$
6	$\overline{M}_1M_2M_4M_3$
6	$M_1M_3M_5M_7M_9M_{11}M_2M_4M_6M_8M_{10}$
5	$M_2M_4M_3M_1$
5	$M_3M_2M_4M_1$
5	$M_1M_3M_5M_7M_9\overline{M}_8\overline{M}_6\overline{M}_4\overline{M}_2$
5	$M_1M_3\ldots M_{15}M_{17}M_2M_4\ldots M_{16}M_{18}$
5	$M_1M_3\ldots M_{19}M_{21}M_2M_4\ldots M_{18}M_{20}$
5	$M_1M_3\ldots M_{25}M_{27}M_2M_4\ldots M_{24}M_{26}$
4	$\overline{M}_1M_2M_4\overline{M}_3$
4	$M_1M_3\overline{M}_2M_4$
4	$M_1\overline{M}_4\overline{M}_2M_3$
4	$M_3M_5M_2M_4M_1$

Continued on next page

Count	Reduced MIC pattern representative
4	$M_1M_3M_5M_7\overline{M}_8\overline{M}_6\overline{M}_4\overline{M}_2$
4	$M_1M_3\ldots M_{13}M_{15}M_2M_4\ldots M_{12}M_{14}$
4	$M_1M_3\ldots M_{19}M_{21}M_2M_4\ldots M_{20}M_{22}$
4	$M_1M_3\ldots M_{21}M_{23}M_2M_4\ldots M_{20}M_{22}$
4	$M_1M_3\ldots M_{23}M_{25}M_2M_4\ldots M_{22}M_{24}$
3	$M_1M_3M_5M_7\overline{M}_6\overline{M}_4\overline{M}_2$
3	$M_2M_4M_6\overline{M}_7\overline{M}_5\overline{M}_3M_1$
3	$M_1M_3M_5M_7M_9M_{11}\overline{M}_{10}\overline{M}_8\overline{M}_6\overline{M}_4\overline{M}_2$
3	$M_1M_3\ldots M_{15}M_{17}M_2M_4\ldots M_{14}M_{16}$
3	$M_1M_3\ldots M_{27}M_{29}M_2M_4\ldots M_{28}M_{30}$
2	$M_1\overline{M}_2\overline{M}_4M_3$
2	$M_1M_3M_5\overline{M}_2\overline{M}_4$
2	$M_2M_4M_3M_5M_1$
2	$M_1\overline{M}_2M_5\overline{M}_3\overline{M}_6M_4$
2	$M_2M_4M_6M_1M_3M_5$
2	$M_3M_5M_2M_4M_6M_1$
2	$M_2M_1M_3\overline{M}_6\overline{M}_7M_5\overline{M}_4$
2	$M_2M_4M_6M_3M_5M_7M_1$
2	$M_4M_6M_2M_5M_7\overline{M}_3\overline{M}_1$
2	$M_2M_4M_6M_8M_1M_3M_5M_7$
2	$M_2M_6M_8M_4M_7M_9\overline{M}_5\overline{M}_3\overline{M}_1$
2	$M_1M_3M_5M_7M_9M_{11}\overline{M}_{12}\overline{M}_{10}\overline{M}_8\overline{M}_6\overline{M}_4\overline{M}_2$
2	$M_2M_4M_6M_8M_{10}M_{12}M_1M_3M_5M_7M_9M_{11}$
2	$M_1M_3M_5M_7M_9M_{11}M_{13}\overline{M}_{12}\overline{M}_{10}\overline{M}_8\overline{M}_6\overline{M}_4\overline{M}_2$
2	$M_1M_3\ldots M_{11}M_{13}M_2M_4\ldots M_{12}M_{14}$
2	$M_1M_3\ldots M_{15}M_{17}\overline{M}_{18}\overline{M}_{16}\ldots\overline{M}_4\overline{M}_2$
2	$M_1M_3\ldots M_{17}M_{19}M_2M_4\ldots M_{16}M_{18}$
2	$M_1M_3\ldots M_{17}M_{19}M_2M_4\ldots M_{18}M_{20}$
2	$M_1M_3M_5M_2M_4\ldots M_{20}M_{22}M_7M_9\ldots M_{21}M_{23}$
2	$M_1M_3\ldots M_{27}M_{29}M_2M_4\ldots M_{26}M_{28}$
2	$M_1M_3\ldots M_{29}M_{31}M_2M_4\ldots M_{28}M_{30}$
2	$\overline{M}_1M_3M_5\ldots M_{19}M_{21}\overline{M}_{37}\overline{M}_{35}\ldots\overline{M}_{25}\overline{M}_{23}M_2M_4\ldots M_{34}M_{36}$
2	$M_1M_3\ldots M_{37}M_{39}\ldots M_2M_4\ldots M_{38}M_{40}$
273	Other

Table 1: Reduced scrambled rearrangement patterns of *O. trifallax* listed in order of frequency.

DOW	Reduced Pattern	Count
123213	$M_1\overline{M}_4\overline{M}_2M_3$	4
	$M_1M_3\overline{M}_2M_4$	4
	$\overline{M}_2M_4M_3M_1$	1
	$\overline{M}_2\overline{M}_4M_3M_1$	1
	$M_1\overline{M}_3M_4M_2$	1
	$\overline{M}_3M_2M_4M_1$	1
	$M_1M_4M_3M_2$	0
	$\overline{M}_1M_4M_3M_2$	0
	$M_1\overline{M}_4M_3M_2$	0
	$\overline{M}_1\overline{M}_4M_3M_2$	0
	$\overline{M}_1M_4\overline{M}_2M_3$	0
	$M_1\overline{M}_3\overline{M}_4M_2$	0
	$M_2M_4M_1\overline{M}_3$	0
	$M_3M_1M_4\overline{M}_2$	0
	$M_2M_4\overline{M}_1\overline{M}_3$	0
	$M_2\overline{M}_4M_1\overline{M}_3$	0
	$M_4M_2\overline{M}_3M_1$	0
	$M_1M_3\overline{M}_2\overline{M}_4$	0
	$M_1\overline{M}_3M_2\overline{M}_4$	0
	$M_1M_4\overline{M}_2M_3$	0
121323	$M_1\overline{M}_2\overline{M}_4M_3$	2
	$\overline{M}_1M_4M_2\overline{M}_3$	1
	$M_2M_1M_4M_3$	1
	$M_1M_4M_2\overline{M}_3$	0
	$\overline{M}_3M_4M_2M_1$	0
	$M_1\overline{M}_2M_4M_3$	0
	$\overline{M}_1\overline{M}_2M_4M_3$	0
	$M_3\overline{M}_2M_4M_1$	0
	$M_1\overline{M}_4M_2\overline{M}_3$	0
	$\overline{M}_1\overline{M}_2M_4M_3$	0
	$M_1\overline{M}_2\overline{M}_4M_3$	0
	$M_1\overline{M}_2\overline{M}_3M_4$	0
	$M_4M_3M_2\overline{M}_1$	0
	$M_4M_3M_2M_1$	0
122133	$\overline{M}_1M_2M_4\overline{M}_3$	4

DOW	Reduced Pattern	Count
123231	$M_2M_4M_3M_1$	5
	$M_1\overline{M}_4M_3\overline{M}_2$	1
	$M_2M_4M_3\overline{M}_1$	1
	$M_2\overline{M}_4M_3M_1$	1
	$M_3M_1M_4M_2$	1
	$M_1M_4M_3\overline{M}_2$	0
	$M_1\overline{M}_3M_4\overline{M}_2$	0
	$\overline{M}_1M_4\overline{M}_3M_2$	0
	$M_2\overline{M}_3M_4M_1$	0
	$\overline{M}_2M_4M_1\overline{M}_3$	0
	$M_3M_1\overline{M}_4M_2$	0
123123	$M_1M_3M_2M_4$	35
	$M_2M_4M_1M_3$	18
	$M_3M_2M_4M_1$	5
	$M_3M_2\overline{M}_4M_1$	1
	$M_1M_4\overline{M}_2\overline{M}_3$	1
	$M_1M_3M_2\overline{M}_4$	0
	$\overline{M}_1M_3M_2\overline{M}_4$	0
	$M_2M_4\overline{M}_1M_3$	0
	$M_2\overline{M}_4\overline{M}_1M_3$	0
	$M_3M_2M_4\overline{M}_1$	0
122313	$M_2M_1M_4\overline{M}_3$	1
	$M_2\overline{M}_1M_4\overline{M}_3$	1
	$M_3\overline{M}_4M_2M_1$	1
	$M_1M_4\overline{M}_3M_2$	0
	$M_1\overline{M}_2M_4\overline{M}_3$	0
	$\overline{M}_1M_4\overline{M}_3M_2$	0
121233	$\overline{M}_1M_2M_4M_3$	6
	$\overline{M}_1M_2\overline{M}_4M_3$	0
	$M_4M_3\overline{M}_2M_1$	0
	$M_1\overline{M}_2M_3\overline{M}_4$	0
123321	$M_1M_3\overline{M}_4\overline{M}_2$	14
	$M_2M_4\overline{M}_3M_1$	1
123312	$M_1M_3\overline{M}_4M_2$	1
122331	$\overline{M}_2M_1M_4\overline{M}_3$	0

Table 2: The number of reduced patterns having four MDSs observed in *O. trifallax*, grouped by their associated double occurrence word.

S3 Nested Repeat-Return Removal Algorithm

```
Get DOW from the contig and store it in RDOW variable
Remove all loops from RDOW
Put RDOW in the ascending order
repeat
    prevWord ← RDOW
    ListOfSets_1 ← empty list, ListOfSets_2 ← empty list
    i ← 1
    while i < length(RDOW) do
        if RDOW[i + 1] = RDOW[i] + 1 then
            letterSet ← {RDOW[i]}
            i ← i + 1
            letterSet ← letterSet ∪ {RDOW[i]}
            while RDOW[i + 1] = RDOW[i] + 1 do
                i ← i + 1
                letterSet ← letterSet ∪ {RDOW[i]}
            end while
            Add letterSet to ListOfSets_1
        else if RDOW[i + 1] = RDOW[i] − 1 then
            letterSet ← {RDOW[i]}
            i ← i + 1
            letterSet ← letterSet ∪ {RDOW[i]}
            while RDOW[i + 1] = RDOW[i] − 1 do
                i ← i + 1
                letterSet ← letterSet ∪ {RDOW[i]}
            end while
            Add letterSet to ListOfSets_2
        end if
        i ← i + 1
    end while
    intersection_1 ← ∅, intersection_2 ← ∅
    for all letterSet_i, letterSet_j in ListOfSets_1 with i ≠ j do
        if |letterSet_i ∩ letterSet_j| ≥ 2 then
            intersection_1 ← intersection_1 ∪ (letterSet_i ∩ letterSet_j)
        end if
    end for
    for all letterSet_i in ListOfSets_1 and letterSet_j in ListOfSets_2 do
        if |letterSet_i ∩ letterSet_j| ≥ 2 then
            intersection_2 ← intersection_2 ∪ (letterSet_i ∩ letterSet_j)
        end if
    end for
    lettersToRemove ← intersection_1 ∪ intersection_2
    for all letter a in RDOW do
```

```
        if a is in lettersToRemove then
            Remove a from RDOW
        end if
    end for
    Remove all loops from RDOW
    Put RDOW in the ascending order
until prevWord = RDOW or RDOW = empty word
```

S4 Proof that the S3 Algorithm Removes Maximal Repeat-Return Words

Let $w = a_1 a_2 \cdots a_{2n}$ be an assembly word. Note that w may be represented uniquely as $w = uv$ where u is a maximal sequence of the form

$$i(i+1)\cdots(i+j-1) \text{ or } i(i-1)\cdots(i-j+1) \tag{$\star$}$$

for some $i, j \in [n]$ and v is a word. Consider the recursive construction where $w_0 = w$ and $w_{k-1} = u_k w_k$ where u_k is the maximal length sequence defined above ($\star$), and w_k is eventually an empty set. This results in a finite set $\{u_1, \ldots, u_p\}$. Define the binary operation $\wedge$ such that, given two words x and y,

$$x \wedge y = \begin{cases} \{a \mid a \in x, a \in y\}, & x \text{ and } y \text{ have at least two letters in common} \\ \emptyset, & \text{otherwise} \end{cases}$$

and the set

$$R = \bigcup_{k,l \in [p], k \neq l} (u_k \wedge u_l).$$

Claim 1. *R is the set of letters that form all maximal sub-repeats and sub-returns of w.*

Proof. Let $x \in R$, then $\exists u_i, u_j$, such that $x \in u_i \wedge u_j$, and we note that $u_i \wedge u_j$ consists of consecutive symbols that correspond to a sub-repeat/sub-return of w. Hence, x is a part of some maximal sub-repeat/sub-return of w.

Let x be a letter of some maximal sub-repeat/sub-return v of w, then there is a letter $y \in v$ such that y is next to x in v. So, $y = x + 1$, or $y = x - 1$ and WLOG assume $y = x + 1$. Consider the first occurrence of v inside w. We claim that x and y belong to the same u_k. Assume otherwise, then we have $x \in u_k$ and $y \in u_{k+1}$. Since $y = x + 1$, then u_k can not be a singleton. Also, u_k can not be of the form $i(i+1)\cdots(x-1)x$, since otherwise u_k can be extended by appending $y = (x+1)$ to u_k, but u_k is maximal. So, x can only belong to u_k of the form $(i)(i-1)\cdots(x+1)x$. Hence, $x + 1 = y \in u_k$, and we also have that $y \in u_{k+1}$, so y already appears twice in w and there is yet another occurrence of y in w. Thus, x and y belong to the same u_k during the first occurrence of v inside w. For the second occurrence of v inside w, we have that either x is before y (i.e. v is a sub-repeat), or y is before x (i.e. v is a sub-return). If x is before y, then the similar argument shows that x and y belong to the same u_l. Assume that y is before x and $y \in u_l$, $x \in u_{l+1}$. Then, since $y = x + 1$, we have that u_l can not be a singleton. Also, u_l can not be of the form $i(i-1)\cdots(y+1)y$, since otherwise u_l can be extended by appending $x = y - 1$ to u_l, but u_l is maximal. Thus, u_l can only be of the form $i(i+1)\cdots(y-1)y$. Hence, $y-1 = x \in u_l$, and $x \in u_{l+1}$, so x appears twice in w as a letter of u_l and u_{l+1}. Also, $x \in u_k$ (which is disjoint from u_l), so x appears 3 times in w. Therefore, x and y belong to the same u_l, so $x \in u_k \wedge u_l \neq \emptyset$, since there are at least two letters (x and y) in u_k and $u_l \Rightarrow x \in R$. □