

## Repetitive DNA Detection and Classification

### **Source**

PILER: identification and classification of genomic repeats, Edgar and Myers  
De novo identification of repeat families in large genomes, Price, Jones, Pevzner

### **Introduction**

#### **Repetitive DNA**

Repetitive DNA refers to substrings of the genome which repeat in which the different instances of the repeat element can have slightly different patterns. Repetitive DNAs are highly prevalent in eukaryotes (About 50% of the human genome is repetitive DNA).

#### **Motivation**

Repetitive DNA is detected for variety of reasons. Among other things, repetitive DNA is generally not found to have any function and thus can be masked out for computational purposes in homology searches to avoid an explosion of unnecessary results. In addition, since repetitive DNA is usually reliably inherited without much change from parents to their offspring, they are suitable for parentage tests.

### **Definition**

#### **Hit**

A **hit** is defined as a local alignment between two regions, **Q** and **T**. **Q** and **T** are called **images** of the hit. **Q = partner(T)** with respect to the hit and vice versa. A hit is completely defined by the end coordinates of **Q** and **T**, referred to as **start(Q)** and **end(Q)**; **start(T)** and **end(T)**.

#### **Dispersed Families (DF)**

A type of repetitive DNA where the substrings are dispersed in the genome is known as a **dispersed family**. This family is often comprised of mobile elements such as Transposons and Retrotransposons. Below is a figure showing the signature induced by a DF on a dot plot of a genome against itself with hits indicated by diagonal lines.

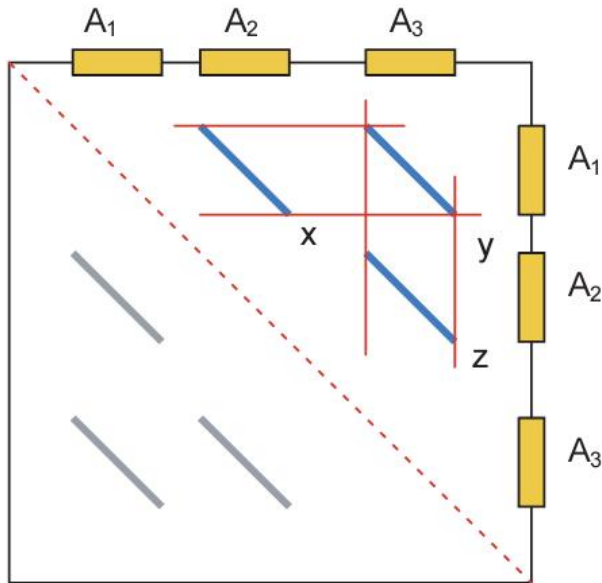


Figure 1 - Signature induced by a DF on a dot plot

In the above figure,  $\text{images}(x) = \{A_1, A_2\}$ ,  $\text{images}(y) = \{A_1, A_3\}$  and  $\text{images}(z) = \{A_2, A_3\}$ .

### Tandem Arrays (TA)

A type of repetitive DNA where substrings are adjacent to one another is known as **tandem arrays**. Tandem arrays induce a pyramidal structure in the dot plot. The repeating elements are known as **satellites**.

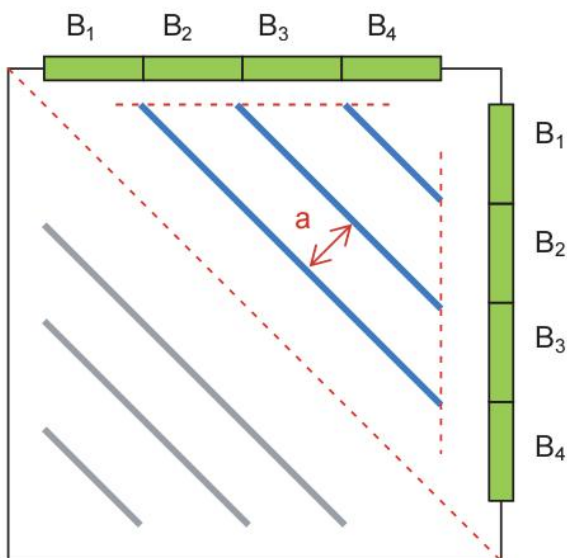


Figure 2 - signature induced by Tandem Array

**Other repeat families**

Pseudo-Satellites (**PS**) are intermediate between Satellites and Dispersed Families. The term **tandem repeat** is often used interchangeably with the term Tandem Array, but for the purpose of this lecture we will make a distinction and define it as images with size 50-2000 bases, separated by 50 to 15000 bases.

We will now discuss the problem of identifying repeat families and the proposed solutions to the problem.

**Problem definition**

The problem of repetitive DNA identification and classification is defined as follows

Given a genomic sequence, identify the repeat families and the positions of their occurrence in the genome.

In this talk we will discuss two papers on the topic. The first one is *PILER: identification and classification of genomic repeats*, by Robert C. Edgar and Eugene W. Myers. The second one is *De novo identification of repeat families in large genomes*, by Price et al.

**PILER****Finding Local alignments**

The first step of the PILER method is to find local alignments. To find local alignments of minimum length  $\lambda$ , and minimum identity  $\mu$ , the authors used the filtration method of Rasumussen et al. in a software tool PALS (Pairwise Alignment of Long Sequences). From this list of hits, a pile is constructed, as described in the next section.

**Constructing Piles**

A **pile** is a list of all hits covering a maximal contiguous region of copy count  $> 0$ . A pile is constructed as follows:

Create vector  $c$  of length  $L$ , set to zero.

Let  $W$  be the set of all  $2N$  images in  $H$ .

For each image  $Q$  in  $W$ :

for  $x = \text{start}(Q)$  to  $\text{end}(Q)$ :

Set  $c[x] = c[x] + 1$

{Now  $c[x]$  is the copy count of base  $x$ }

```

Set P =0 {P is number of piles found so far}.

For x =1 to L:
  if c[x-1] = 0 and c[x] > 0: {is x start of new pile?}
    Set P =P + 1
  if c[x] > 0:
    Set c[x] =P

{Now c[x] is the identifier of the pile that covers base x,
or zero if x is unique }

Create P empty piles. A pile is a list of images.

For each image Q in W:
  Set p =c[start(Q)]
  Add Q to pile p.

```

Assuming a constant average hit length, the above procedure is  $O(N)$  and the authors claim it is efficient in practice for typical input data.

### PILER-DF

PILER-DF is "...a search method designed to find intact, isolated members of a dispersed family" which are identified as sets of  $t$  or more globally alignable piles. They define the parameter  $t$  as being greater than three to distinguish a DF from a segmental duplication in the genome. **pile(Q)** is defined to be the pile containing **Q**, and **is-global-image(Q)** is true if **Q** covers a fraction  $\geq g$  of the bases in **pile(Q)**, where  $g$  is another parameter smaller than 1.

```

Let G be a graph with one node for each pile and no edges.

```

```

Is-global-image(Q) = true if
  # bases in Q >= g * ( # bases in pile(Q) )

```

```

For each pile p in P:
  For each image Q in p:
    Let T = partner(Q)

```

```

if is-global-image(Q) and is-global-image(T) :
    Add edge p-pile(T) to G

```

```

Find connected components of G of order >= t

```

At the end of the algorithm, each connected component is classified as a dispersed family and can be interpreted as a putative intact mobile element. By requiring piles rather than hits to align globally, they avoid the problem of multiple alignment between intact instances of element **A** and fragments of **A**, **a**, producing a consensus similar to **a** rather than the intact element of **A**.

### PILER-PS

The problem of identifying pseudo satellites is very similar to identifying dispersed families. Thus, the PILER-PS algorithm is identical to PILER-DF, with the exception that hits are identified by a banded search as opposed to a full search for self-alignments. Using a banded search enforces the requirement that PSs are clustered, and it also allows for a faster and more sensitive search for hits, possibly enabling the discovery of more highly degraded instances. It has the added benefit of reducing noise produced by alignments to more distant repeats in other regions of the genome. This in turn may result in cleaner piles and increased sensitivity.

### PILER-TA

The TA derivative of PILER is a method to search for Tandem Arrays, as identified as pyramids in the dot plot. Recall that Tandem Arrays have pyramids as their signature. Also note that hits in a pyramid belong to the same pile. Thus, by requiring that images are separated by a distance of at most beta, we can avoid considering every pair of hits by using a banded search.

Before discussing the PILER-TA algorithm we will define two terms:

1. **first(h)**: image in h with the smaller start coordinate.
2. **last(h)**: image in h with the larger start coordinate.

With this definition, we will discuss the PILER-TA algorithm:

```

For each pile p:
    Create an empty graph G with all hits in the pile
    For each pair of hits (h, h') in p:
        Set shorter_length = min(|h|, |h'|)
        Set longer_length = max(|h|, |h'|)
        Set Q1 = first(h) // B1,B2,B3 in figure 2
        Set T1 = last(h) // B2,B3,B4 in figure 2

```

```

Set Q2 = first(h') // B1,B2 in figure 2
Set T2 = last(h') // B3,B4 in figure 2
Set dS = (start(Q)-start(Q')) / shorter_length //0-0=0
Set dE = (end(T)-end(T')) / shorter_length //4-4=0
if shorter_length / longer_length > 0.5 and
    |dS| < m and |dT| < m:
    Add edge h-h' to G
Find connected components of G

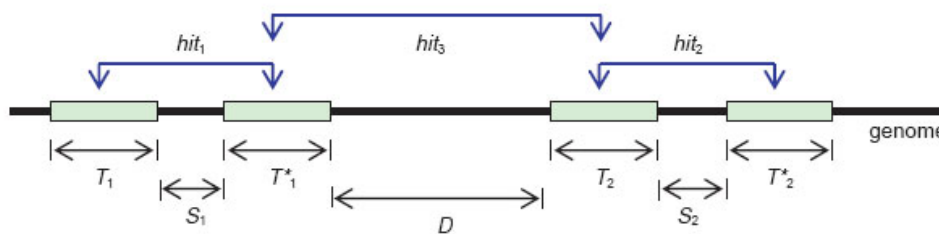
```

At the end of the iteration, each connected components of  $G$  is a Tandem Array. The parameter  $m$ , a constant between 0 and 1, determines how closely the endpoints must align, expressed as a fraction of the length of the shorter hit.

### PILER-TR

PILER-TR is used to identify Tandem Repeat families in the genome. In the pre-processing stage, candidate TRs are identified as images within a length that's typical of a TR (50 to 2000 bases) and separated by a distance reasonable for TRs (50 to 15000 bases). Because Satellites and PSs often induce hits meeting the criteria, they are identified and masked in this stage. Once the preprocessing step is done, TR candidates are found by using the following two passes:

1. Perform a banded search for TR candidates.
2. Find hits that align TR pairs to each other.



“In the first pass, candidate TRs are identified as hits such as  $hit_1$  and  $hit_2$  that align images of lengths ( $T$ ) and separations ( $S$ ) within bounds for mobile elements with TRs. In a second pass, hits that align candidates with each other, such as  $hit_3$ , are identified by the following criteria: 1) its images are approximately a pair of candidate TRs ( $T * 1$ ,  $T_2$ ) from different hits found in the first pass, 2) the candidates ‘joined’ by this hit have similar separations ( $S_1$ ,  $S_2$ ) and (3) its images are sufficiently separated in the genome, i.e. have large enough  $D$ .”

At the end of the algorithm connected components are identified and interpreted as putative families of TR elements.

## Library Construction

A complete genome annotation requires constructing a library of elements for use by a separate tool such as BLAST (Altschul et al., 1990) or RepeatMasker. This is because most instances of mobile elements in currently sequenced model organisms are either not intact or not isolated and thus not directly reported by PILER-DF. Thus, the authors used MUSCLE (Edgar, 2004) to create multiple alignments of family members found by PILER, from which consensus or centroid sequences are produced. The output is then manually curated by screening for nucleotide and protein similarity to known repeats and known functional elements. The resulting library can then be used by BLAST or RepeatMasker to find intact and partial instances of repeat families.

## Results

Edgar et al validated PILER-DF on model organisms including *D.melanogaster* and *A.thalania*, finding many known mobile elements and a few that had not been previously reported. They found that PILER-DF has high specificity, with sensitivity that varies depending on the genome.

PILER-TA identified 67 TA of motifs ranging in length from 61 to 1950 bases in the *H.sapiens* chromosome 1. PILER-PS identified 55 PSs of length ranging from 113 to 1518 bases. In some cases, PILER misidentified the length of the repeat, reporting two or more concatenated instances of the true motif.

Finally, the authors ran PILER-TR on the *D.melanogaster* genome which included heterochromatin contigs. Using the search, they identified twenty-four TR families, several of which appeared to be incompletely masked satellites or PSs.

This concludes the section on PILER, and we will now focus our attention to the paper by Price et al.

## ***De novo identification of repeat families in large genomes***

### RepeatScout

The RepeatScout algorithm as discussed in the paper improves on the RECON algorithm by Bao and Eddy, 2002. The algorithm builds repeat families using high-frequency L-mers as seeds. The input to RepeatScout are DNA sequences  $S_1, S_2, \dots, S_N$ , each of which contains similar repeat elements and extends past the repeat elements on either side. The output is the substring  $R_1, R_2, \dots, R_N$  which give the repeat element boundaries and the consensus sequence  $Q$ .  $Q$  is defined to be the sequence that maximizes the following:

$$A(Q; S_1, \dots, S_n) = \left[ \sum_k \max\{a(Q, S_k), 0\} \right] - c|Q|$$

where  $\mathbf{a}(\mathbf{Q}, \mathbf{S}_k)$  is a pairwise fit-preferred alignment score and  $|\mathbf{Q}|$  is the length of  $\mathbf{Q}$ . The repeat frequency threshold  $\mathbf{c}$  imposes a minimum threshold on the number of repeat elements which must align with any given position of  $\mathbf{Q}$ .

For the  $\mathbf{a}(\mathbf{Q}, \mathbf{S}_k)$  term, a fit-preferred alignment score is used. This leads to a consensus sequence  $\mathbf{Q}$  with proper repeat boundaries.

$$\begin{aligned}
 f(i, 0) &= \max(-\gamma i, -p), \\
 f(0, j) &= 0, \\
 f(i, j) &= \max \begin{cases} f(i-1, j-1) + \mu_{ij} \\ f(i, j-1) - \gamma \\ f(i-1, j) - \gamma \\ -p \end{cases}, \\
 a(Q, S) &= \max_{i,j} \begin{cases} f(i, j) & \text{if } i = |Q| \\ f(i, j) - p & \text{if } i < |Q| \end{cases}.
 \end{aligned}$$

**Figure 3 - Fit preferred alignment score**

### Optimizing $\mathbf{A}(\mathbf{Q}; \mathbf{S}_1, \dots, \mathbf{S}_n)$

Now that the objective function  $\mathbf{A}(\mathbf{Q}; \mathbf{S}_1, \dots, \mathbf{S}_n)$  is defined, the next step is to optimize it. The naïve approach of solving an n-dimensional dynamic programming is intractable. Thus, the authors propose a heuristic approach of extending short exact seeds to identify high-scoring alignments, ala BLAST (Althchul et al, 1990) as follows.

Given a high-frequency **I-mer**, let  $\mathbf{S}_1, \dots, \mathbf{S}_n$  be the regions surrounding exact matches of the **I-mer**, extended far enough on each side (say 10K bp) such that they extend past the boundaries of the repeat elements in question. Set  $\mathbf{Q}_0$  equal to the **I-mer** and greedily extend the consensus sequence  $\mathbf{Q}$  maximizing the objective function one nucleotide at a time: at iteration  $\mathbf{t}$ , compute  $\mathbf{A}(\mathbf{Q}_t.\text{append}(\mathbf{N}); \mathbf{S}_1, \dots, \mathbf{S}_n)$  for each nucleotide  $\mathbf{N}$  in  $\{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}$  and set  $\mathbf{Q}_{t+1}$  equal to the choice of  $\mathbf{Q}_t.\text{append}(\mathbf{N})$  which maximizes  $\mathbf{A}(\mathbf{Q}_t.\text{append}(\mathbf{N}); \mathbf{S}_1, \dots, \mathbf{S}_n)$ . Note that alignment scores  $\mathbf{a}(\mathbf{Q}_t.\text{append}(\mathbf{N}), \mathbf{S}_k)$  can be computed quickly from caching the alignment scores of the previous iteration. In this way,  $\mathbf{Q}$  is greedily extended one nucleotide at a time to a progressively longer consensus sequence. The extension process terminates after a specified number of iterations,  $\mathbf{I}$ , fails to improve on the optimal score  $\mathbf{A}(\mathbf{Q}; \mathbf{S}_1, \dots, \mathbf{S}_n)$ . The original **I-mer** seed  $\mathbf{Q}_0$  is extended to the right and then to the left in this manner, thus resulting in the consensus sequence optimizing  $\mathbf{A}(\mathbf{Q}; \mathbf{S}_1, \dots, \mathbf{S}_n)$ . If the consensus sequence obtained in this fashion is shorter than a specified length (say 50 bp), it is discarded in order to prevent spurious high-frequency **I-mers** from entering the repeat family library.

The above procedure outputs a single repeat family consensus sequence  $\mathbf{Q}$  for a given genome and a high-frequency **I-mer**. In applying the procedure to all high-frequency **I-mers** to identify all repeat families in a genome, occurrences of a repeat family after its consensus sequence  $\mathbf{Q}$  is computed is identified and **I-**

**mer** frequency count is adjusted to exclude counts from those repeat occurrences in order to preclude rediscovering the same repeat family. Thus, the algorithm is as follows:

1. Build a table of high-frequency l-mers
2. Extend the most frequent l-mer to a repeat family Q
3. Identify occurrences of Q in the genome and adjust l-mer frequency counts to exclude counts from occurrences of Q
4. Repeat on the most frequently remaining l-mer

The algorithm terminates when there are no remaining **l-mers** with frequency at least **m**, a fixed frequency threshold.

## Results

RepeatScout was benchmarked using the *C.briggsae* genome (Stein et al, 2003), a recently assembled genome of length 108Mb, which has already been analyzed using RECON. RepeatScout was able to find 1391 repeat families of total length 0.65Mb, where RECON found 723 repeat families of total length 0.43Mb.

Price et al then applied RepeatScout to the human, mouse, and rat X chromosomes with the goal of identifying repetitive sequence which excludes low-complexity or tandem repeats which is not already annotated. After manually curating the output from RepeatScout, they compared the number of repeat families in the resulting library and their total length to the Repbase Update library of transposon families (Jurka, 1998, 2000), an existing library of known repeats. The following table compares the runs of RepeatMasker using either the RepeatScout library or the Repbase library:

	Masked by RepeatScout (Mb)	Not masked by RepeatScout (Mb)	Total (Mb)
Human X chromosome			
Masked by Repbase	51.7	7.2	58.9
Not masked by Repbase	2.9	62.0	64.9
Total	54.6	69.2	123.8
Mouse X chromosome			
Masked by Repbase	43.2	2.7	45.9
Not masked by Repbase	6.4	62.4	68.7
Total	49.6	65.1	114.6
Rat X chromosome			
Masked by Repbase	49.8	3.2	53.0
Not masked by Repbase	7.2	77.3	84.5
Total	56.9	80.6	137.5

The authors concluded that 2.9Mb (2%) of the human X chromosome, 6.4Mb(4%) of the mouse X chromosome and 7.2Mb (4%) of the rat X chromosome consist of previously un-annotated repetitive sequence. Price et al conjectured that since rodent genomes were assembled recently, the manually curated Repbase library of rodent transposon families was incomplete—and that it explains both the relatively large amount of repetitive sequence identified by RepeatScout alone and the relatively small amount of sequence identified by Repbase alone in the mouse and rat X chromosomes. Since the Repbase library of human transposon families is based on many years of manual curation, the authors think that the repetitive sequence identified by RepeatScout alone consists of diverged duplication units which are not presently annotated as segmental duplications.

### ***Conclusion***

In this talk, we discussed two algorithms which address the problem of DNA repeats, PILER and the RepeatScout algorithm. PILER focuses more on finding diverse repeat families and uses MUSCLE to find the consensus sequence, whereas RepeatScout focuses more on finding the consensus sequence given members of a repeat family.