



Institut Pasteur

Characterization of Genome Coverage and Identification of Genomic Regions of Interest (ROIs)

Thomas Cokelaer, Dimitri Desvillechabrol

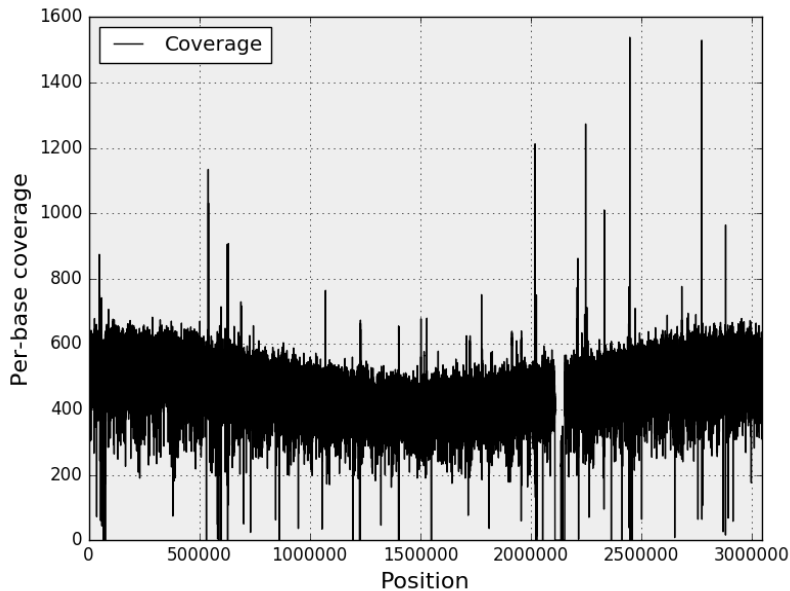
Jan 24th 2017, Institut Pasteur (HUB talk)

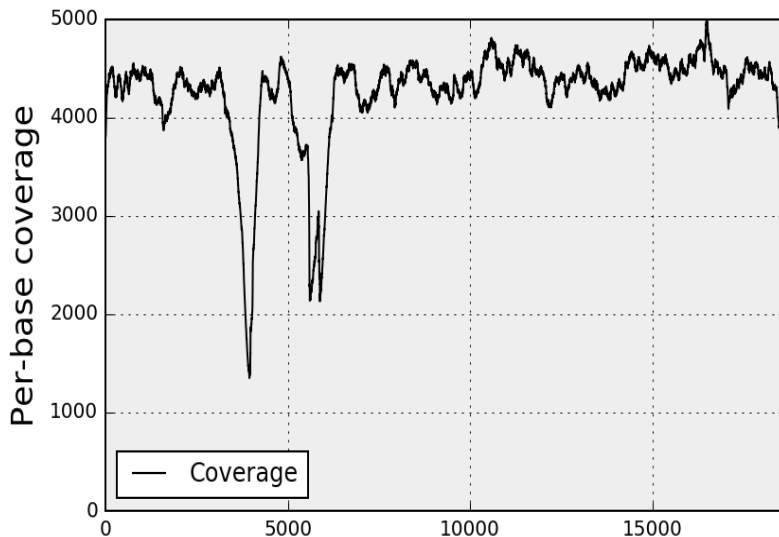
Definition: The number of reads mapped to a specific position, b , within the reference genome.

Notation: $C(b)$ also denoted C_b

Theoretical distribution: Poisson distribution but in practice over dispersed. The poisson parameter is distributed according to a Gamma hence leading to a negative binomial (See e.g., Linder et al 2013).

Bacteria case (low/high μ components and del. region)





Question: how to automatically detect and characterise under and over covered genomic regions

1. The algorithm
 1. Detrending (running median)
 2. Mixture model estimation (Gaussian approximation)
 3. Set a statistics (z-score)
 4. Clustering (double threshold)
2. Demo
3. Applications

1. Detrending

We divide C_b by its moving average (MA), or even better its running median (RM) defined as

$$RM_W(b) = \text{median}(C(b - V), \dots, C(b + V))$$

W is the running window and $V = (W - 1)/2$.

The normalised genome coverage

$$\tilde{C}_b = \frac{C_b}{RM_W(b)}$$

1. Detrending

We divide C_b by its moving average (MA), or even better its running median (RM) defined as

$$RM_W(b) = \text{median}(C(b - V), \dots, C(b + V))$$

W is the running window and $V = (W - 1)/2$.

The normalised genome coverage

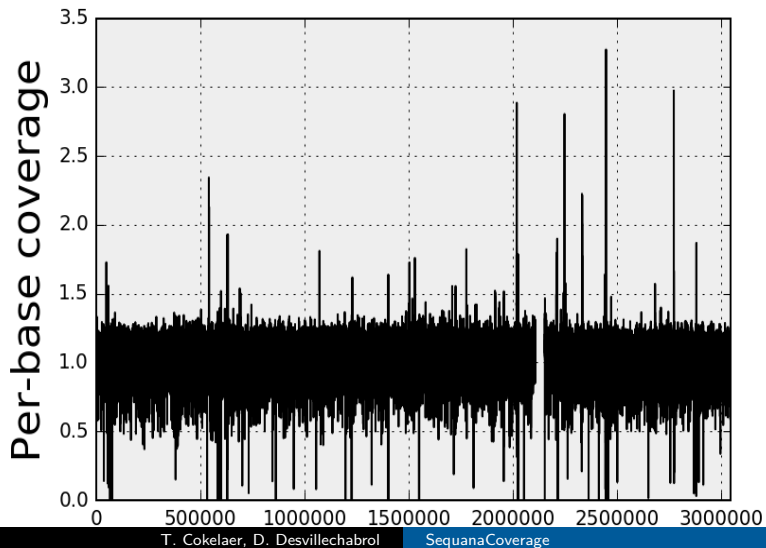
$$\tilde{C}_b = \frac{C_b}{RM_W(b)}$$

Computational note

Running median is slow due to the sorting task.

Solution: a rolling window + a bisection method to insert new element in a sorted list + efficient insert/deletion in a list (B-tree) helps: **Only a few seconds to scan a 3Mb-length genome.**

Normalised coverage



2. Building a statistics

Definition: the normalised data can be decomposed into a **central** distribution \tilde{C}^0 and a set of **outliers** \tilde{C}^1 (above and below the central distribution)

$$\tilde{C}_b = \left\{ \tilde{C}_b^0, \{ \tilde{C}_b^+, \tilde{C}_b^- \} \right\}$$

Hypothesis 1: Central distribution is predominant.

$$\left| \tilde{C}_b^0 \right| > \left| \tilde{C}_b^1 \right|$$

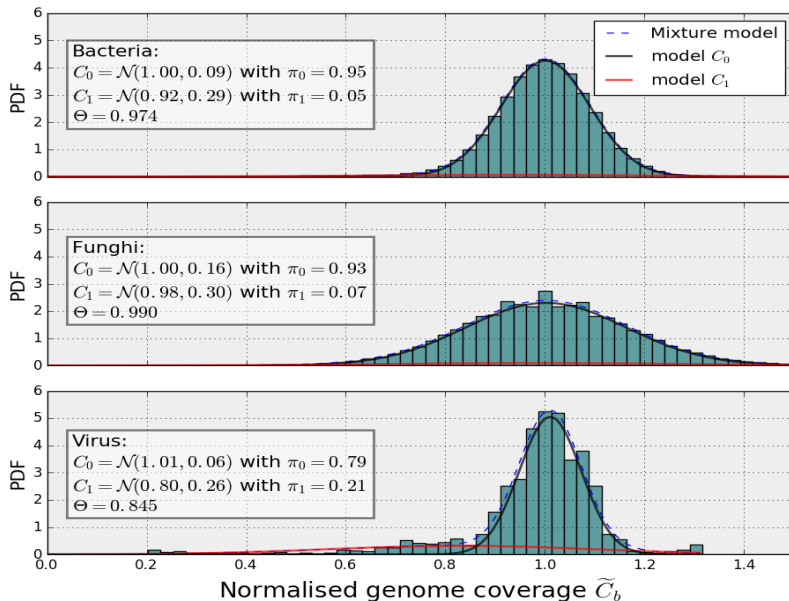
Hypothesis 2: The normalised genome coverage follows a Gaussian distribution in particular the central distribution

$$PDF(\tilde{C}_b^0) \sim \mathcal{N}(\mu_0, \sigma_0)$$

We consider regime with $\delta \gg 1$. However, the algorithm works for low values: $\delta \sim 5 - 10$.

The central distribution C_0 will be fitted to a Gaussian distribution while the outliers will be fitted to another distribution.

- With an EM algorithm using $k = 2$ distributions we can estimate the parameters.
- The parameters of the outliers components are not used.
- The average of the central distribution is 1 (by definition)
- Note that the average of the outliers can be around 1 as well if the weights of C_+ and C_- are equivalent



C. From a constant to adaptative z-score

Assuming that the central distribution is the null hypothesis, we can now assign a **z-score in the normalised space**:

$$z(b) = \frac{\tilde{C}(b) - \tilde{\mu}_0}{\tilde{\sigma}_0}$$

We can replace \tilde{C}_b by its expression ($C_b/RM_W(b)$) and express C_b as a function of the running median, the $z(b)$ and the parameters of the central distribution:

$$C(b) = (\tilde{\mu}_0 + z(b)\tilde{\sigma}_0) RM_W(b).$$

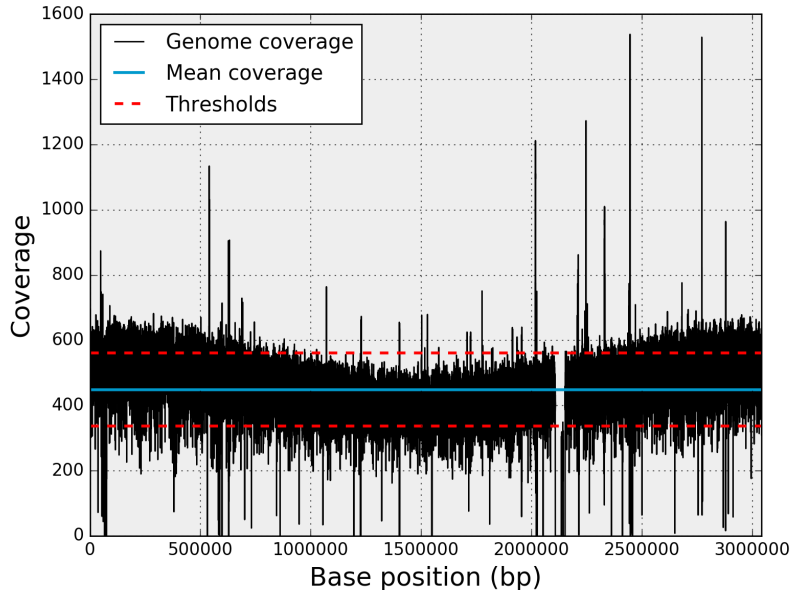
C. From a constant to adaptative z-score

We can now set a constant threshold in the normalised space (e.g. $\lambda \pm 3$) and get an adaptative threshold in the original space.

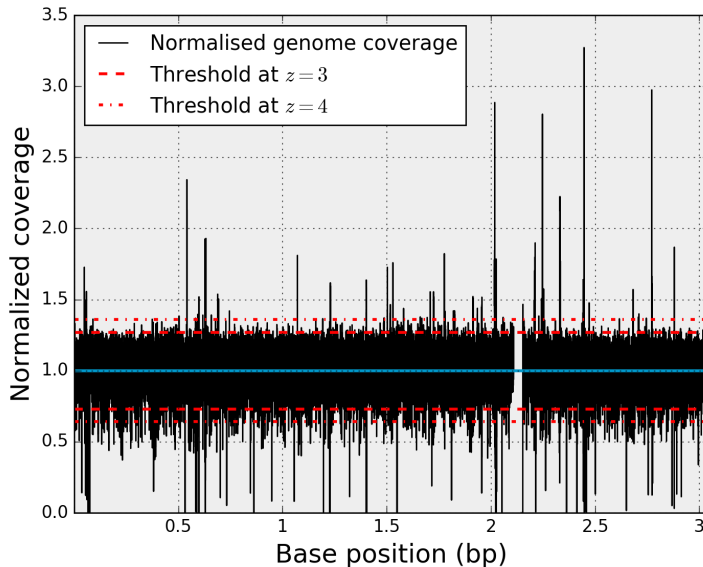
We can derive an **adaptative threshold** in the original space that is function of the genome position:

$$\tilde{\delta}^{\pm}(b) = (\tilde{\mu}_0 \pm \lambda^{\pm} \times \tilde{\sigma}_0) RM_W(b). \quad (1)$$

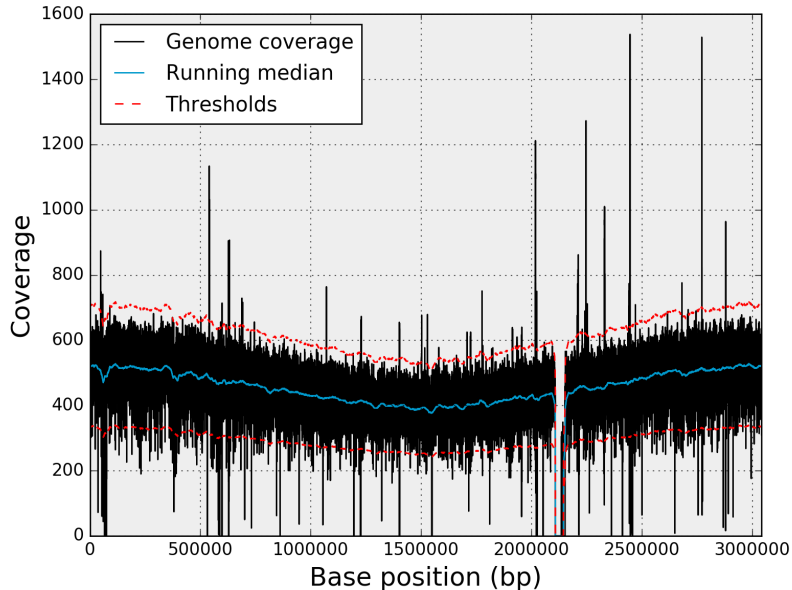
constant thresholds in the original space



constant thresholds in the normalised space



adaptative thresholds in the original space



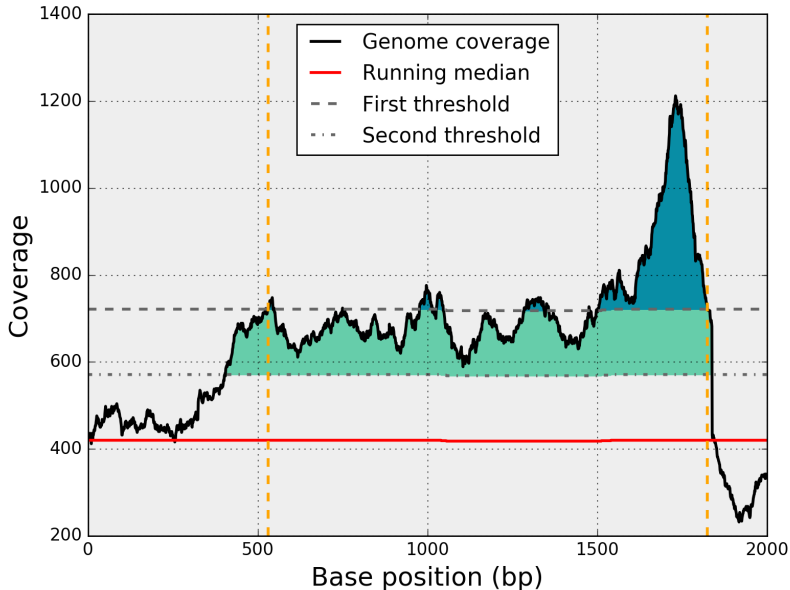
D. Regions of interest (clustering)

On a 3Mb genome with low thresholds the number of outliers are high. $\lambda = 3$ means about 4000 events by pure chance.

We need a strategy to reduce the number of interesting events. This is achieved by clustering the data.

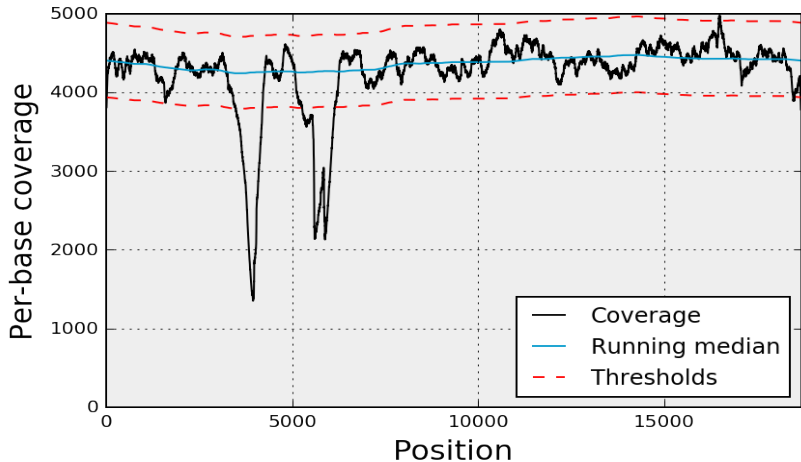
Besides, to cluster close-by events further, we proceed with a double-thresholds approach.

Double thresholds



DEMO

```
sequana_coverage --download-genbank JB409847  
sequana_coverage --input JB409847.bed -w 3001 --circular  
--genbank JB409847.gbk --show-html
```



- A robust and fast algo. to detect under/over covered regions
- The algorithm is made of 3 steps:
 1. Normalisation (running median)
 2. Set a statistics using EM to estimate mixture model
 3. Clustering of events in original space
- Implementation in Sequana as a standalone
 - HTML reports with ROIs provided as sortable tables
 - Identify repeated regions
 - A genbank can be provided to annotate the report

Dimitri Desvillechabrol, Christiane Bouchier, Sean Kennedy, Thomas Cokelaer
Detection and characterization of low and high genome coverage regions using an efficient running median and a double threshold approach. bioRxiv 092478;
doi: <http://dx.doi.org/10.1101/092478>