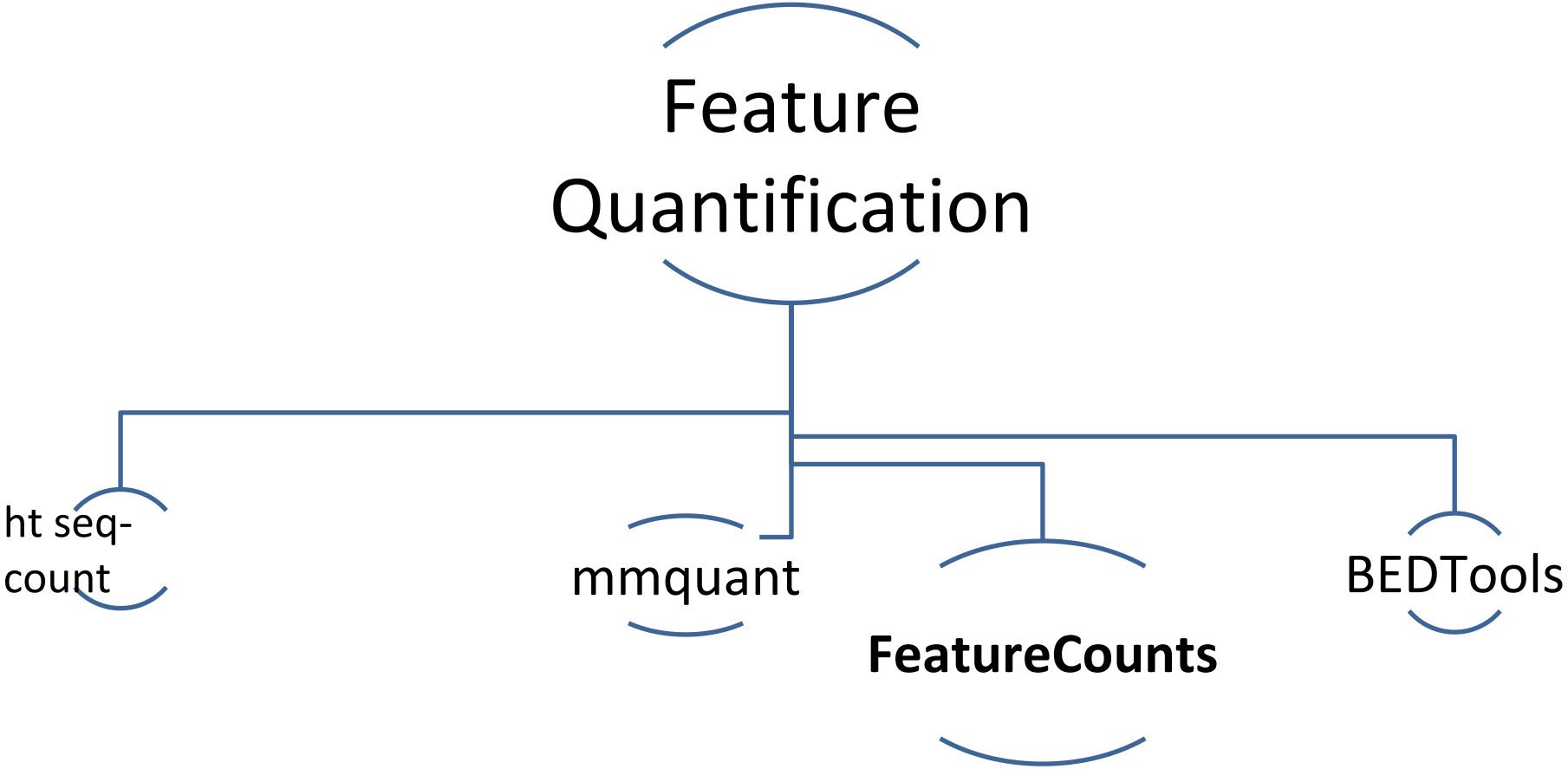
MASARYKOVA UNIVERZITA

FeatureCounts

Ashebir Gogile
Uco540613


Feature Quantification
ht seq-count
mmquant
FeatureCounts
BEDTools

National Library of Medicine
National Center for Biotechnology Information

Log in

PubMed.gov

Search

Advanced

User Guide

Save | Email | Send to | Display options

› Bioinformatics. 2014 Apr 1;30(7):923-30. doi: 10.1093/bioinformatics/btt656. Epub 2013 Nov 13.

featureCounts: an efficient general purpose program for assigning sequence reads to genomic features

Yang Liao [1], Gordon K Smyth, Wei Shi

Affiliations + expand

PMID: 24227677 DOI: 10.1093/bioinformatics/btt656

FULL TEXT LINKS

OXFORD ACADEMIC

deepdyve

ACTIONS

Cite

Collections

- NGS technologies generate millions of short sequence reads,

  which are usually aligned to a reference genome.

- For many downstream analysis is the number of reads mapping to each genomic feature(exons,genes)

- The process of counting reads is called read summarization.

- ***featureCounts*:**
  - ✓ an ultrafast and accurate read summarization program
  - ✓requires far less computer memory.

- a highly efficient general-purpose read summarization program that counts mapped reads for genomic features

- It can be used to count both RNA-seq and genomic DNA-seq reads (SAM/BAM files)

- It works with either single or paired-end reads

featureCounts

- uses genomics annotations in GTF or SAF format for counting genomic features (exons) and meta-features (genes).

- When you want to analyze the data for differential gene expression analysis, it would be convenient to have counts for all samples in a single file (gene count matrix).
- Gene count matrix file run featureCounts on all mapped files at once.

```
# meta-feature (gene) level count
featureCounts -t 'exon' -g 'gene_id' -a annotation.gtf -T 10 -o counts.txt library1.bam library2.bam
# use -f option for feature (exon) level count
```

- But, when you run a featureCounts for large samples individually, then the counts for each sample will be in a separate text file.

- To get the merged gene count matrix from all individual counts files, you can use bioinfokit v2.0.5

```
# run this Python code (in a Python interpreter) from a folder where all files are present
from bioinfokit.analys import HtsAna
# make sure all individual count files are present in same folder
# by default, it assumes each count file has .txt extension
HtsAna.merge_featureCount()
```

Input and output

Inputs

- takes as input Sequence Alignment(SAM)/Binary Alignment(BAM) files and

- an annotation file including chromosomal coordinates of features.

- The annotation file should be in either GTF format or a simplified annotation format (SAF) as shown below:

- GeneID Chr Start End Strand
- 497097 chr1 3204563 3207049 -
- 497097 chr1 3411783 3411982 -
- 497097 chr1 3660633 3661579 -

outputs

- are numbers of reads assigned to features (meta-features).

- stat info for the overall summrization results,(no of successfully assigned reads and no of reads that failed to be assigned due to various reasons

- you can see the output file gene_matrix_count.csv in the same folder, which has counts merged for all samples.

```
# gene_matrix_count.csv
Geneid,sample1.bam,sample2.bam,sample3.bam
PGSC0003DMG400015133,0,7,2
PGSC0003DMG400015132,72,95,155
PGSC0003DMG400022764,42,78,77
PGSC0003DMG400022799,2,3,5
```

ALGORITHM

- **Overlap of reads with features**:

FeatureCounts takes account of any gaps (Indels, exon–exon junctions) that are found in the read.

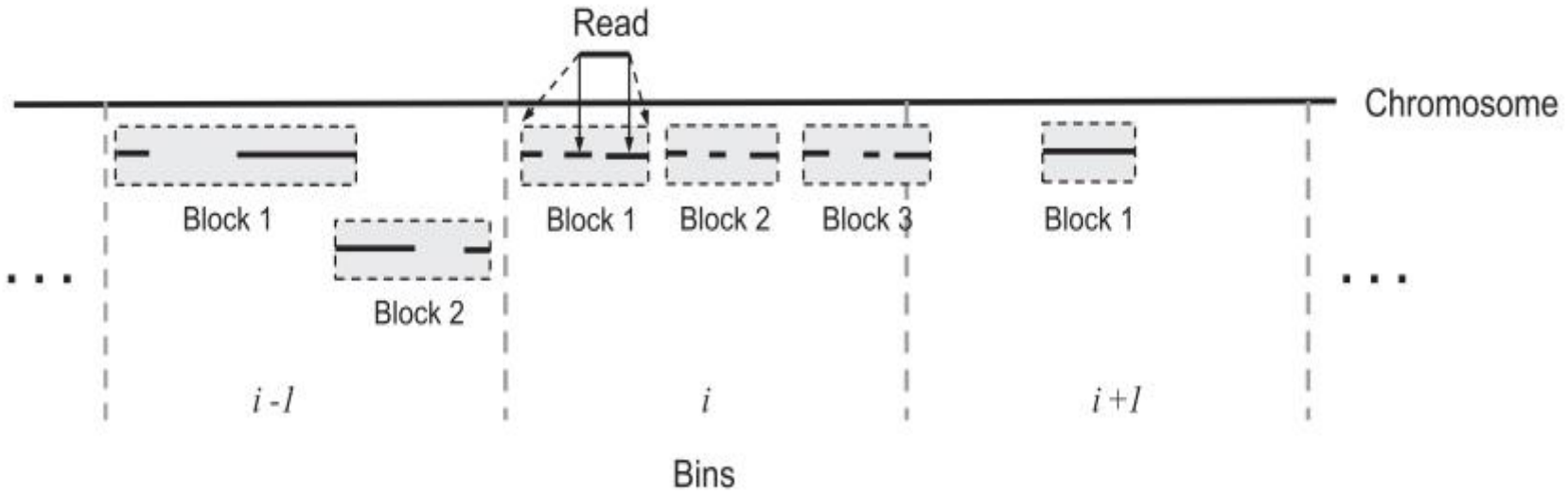
- **Multiple overlaps:**

featureCounts provides users with the

option to either exclude multi-overlap reads or to count them for each feature that is overlapped.

- **Chromosome hashing**: used to generate a hash table for the reference sequence names.
- matching reads and features by reference sequence

- **Genome bins and feature blocks**:A two-level hierarchy is created for each reference sequence.



The use of a hierarchical data structure (features within blocks within bins) is a key component of the featureCounts algorithm.

- The query read

compared first with genomics bins,

with feature blocks within within bins

then features in any overlapping blocks.