Base-quality trimming

Concepts and tools

Recap

- A data set of sequence reads needs quality-control
- There are various types of quality issues to consider
- **Base-quality** is just one of them
- Others include chimaeras, artefactual duplicates etc
- See earlier slideshows for a reminder of:
  - what base-quality is
  - how it is determined – what those **quality scores** mean
  - considerations of some basic statistics of base-quality
  - some tools for viewing these statistics (e.g. FASTQC)

Quality scores in context (1)

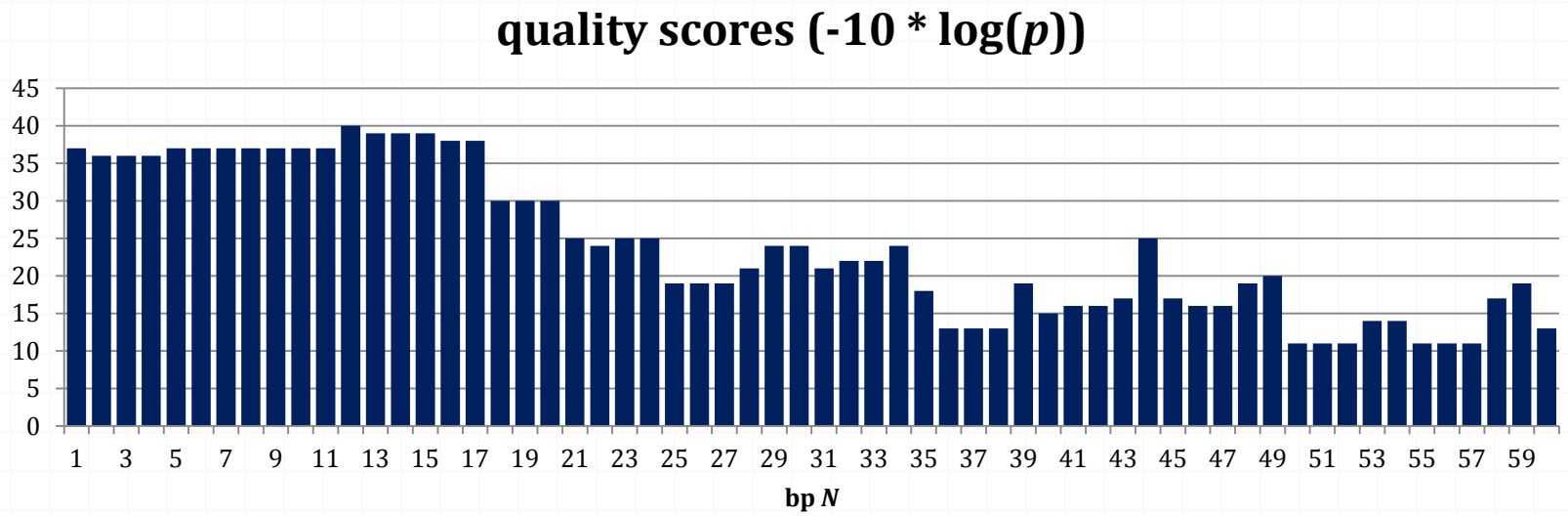
- Remember that a quality score (phred score) indicates the probability of the base-call being correct
- It is useful to consider how quality scores relate to the type of sequencing performed
- Some benchmarking exercises assessing various trimming tools may consider them in one context
  - e.g. effect on RNA-seq mapping efficacy

Quality scores in context (2)

- How damaging would miscalled bases be:
  - at either end of RNA-seq reads?
    - affects whether (or where) the read is mapped
  - In genome sequencing with very high coverage?
  - At the ends of 16S amplicons?
    - i.e. in non-variable regions
  - In the middle of 16S amplicons?
  - In shotgun metagenomic reads?

Example – a single (very short) sequence read

- Quality scores (phred scores) look like this:



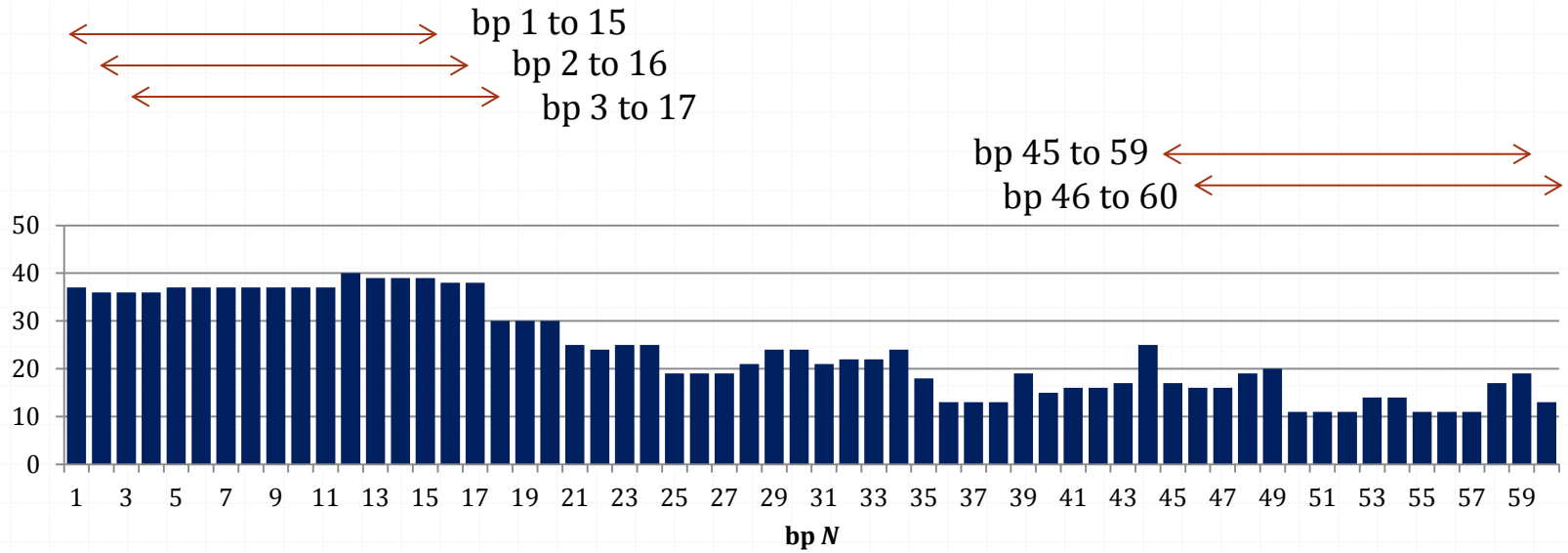
- it's really not a very good read
- how might it best be quality-trimmed?

Approaches to quality-trimming

- Broadly – there are two types of approach
  1. Window-based
  2. Running sum-based
  3. Other rule-based
- In types 1 and 2, the basic approach may be supplemented with other rules, such as:
  - the permitted minimum length of read, after trimming
- Also, trimming might be applied to:
  - the 3'-end only
  - the 5'-end only
  - both

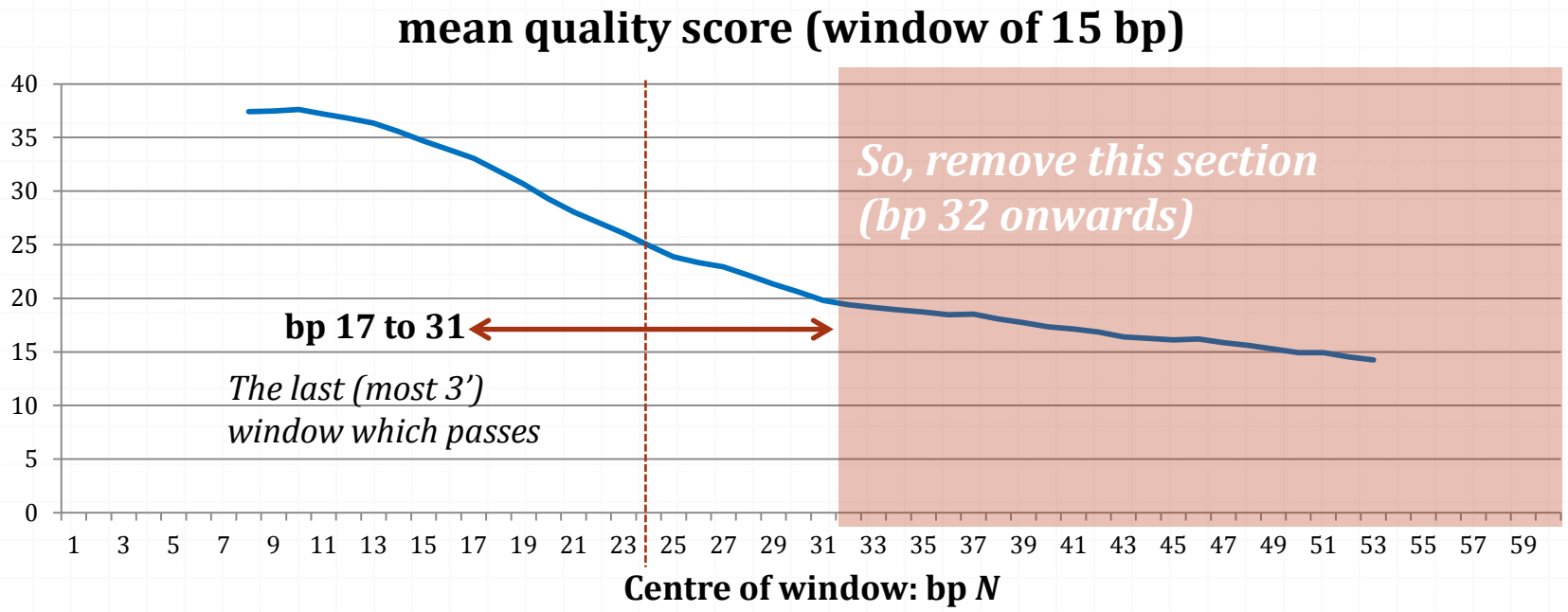
Window-based quality-trimming

- The "sliding window" – a simple concept, used a lot in many types of sequence analysis
- E.g. a window of 15bp:



Window-based quality-trimming

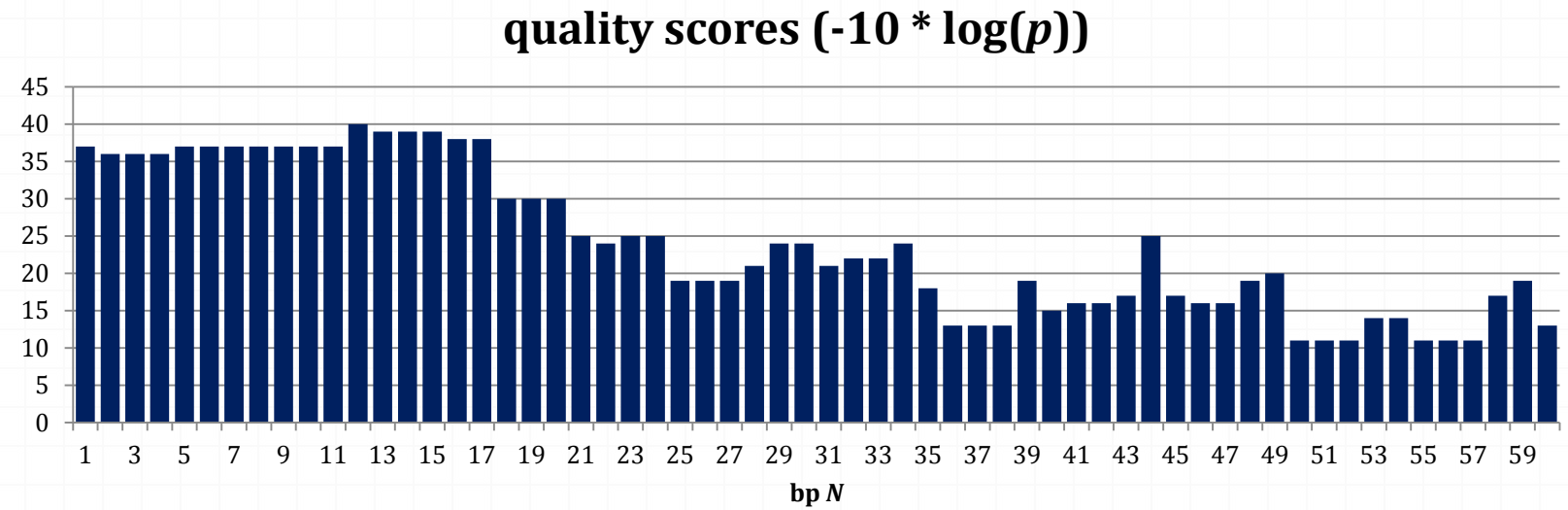
- Calculate a property of each window, e.g. **mean quality score**
- Cut when the property fails to meet a quality criterion
  - e.g. mean **quality score must be ≥ 25**

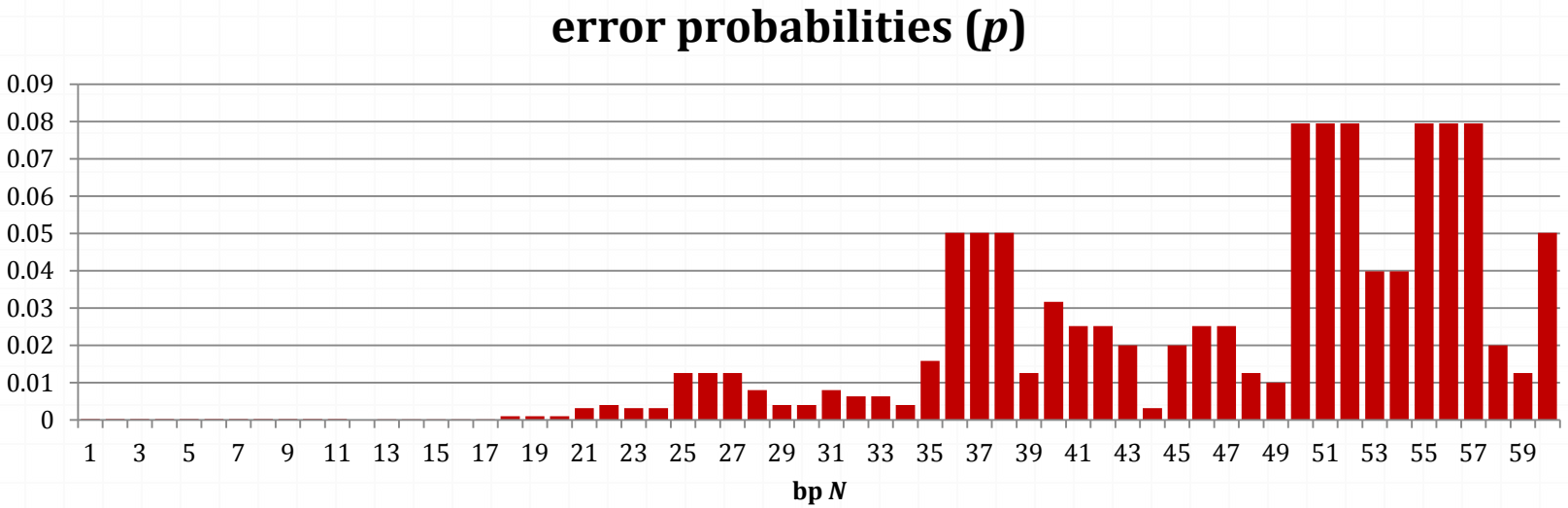


“Running-sum” methods

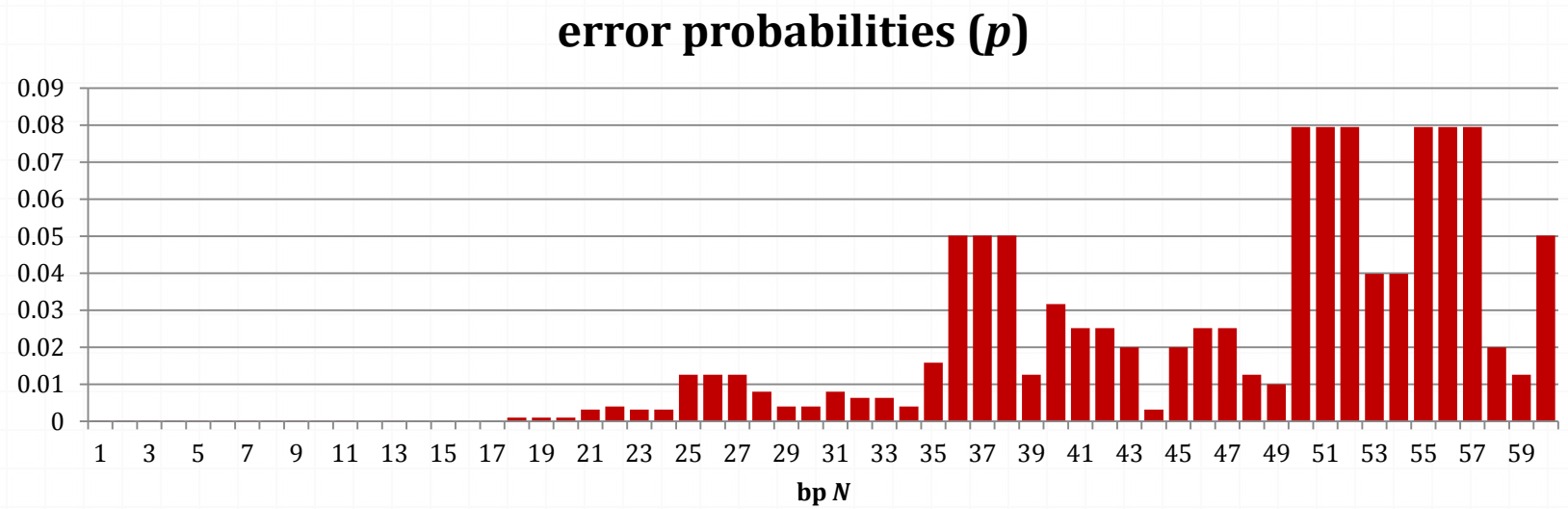
- Example – modified Mott algorithm, as used in Phred/Phrap and CLC Workbench
  - Note that this does **not** use the phred quality-scores (Q scores)
  - It uses the underlying phred **error probability values**
  - Some of the literature on this is very confusing!

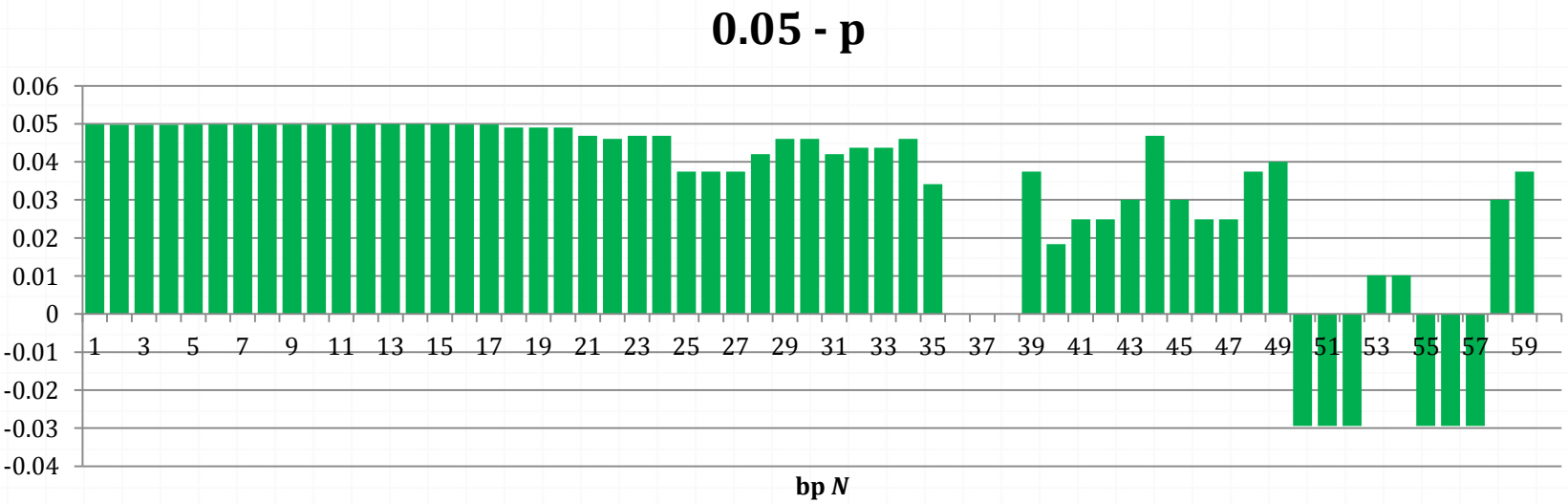
- Using the same example read:



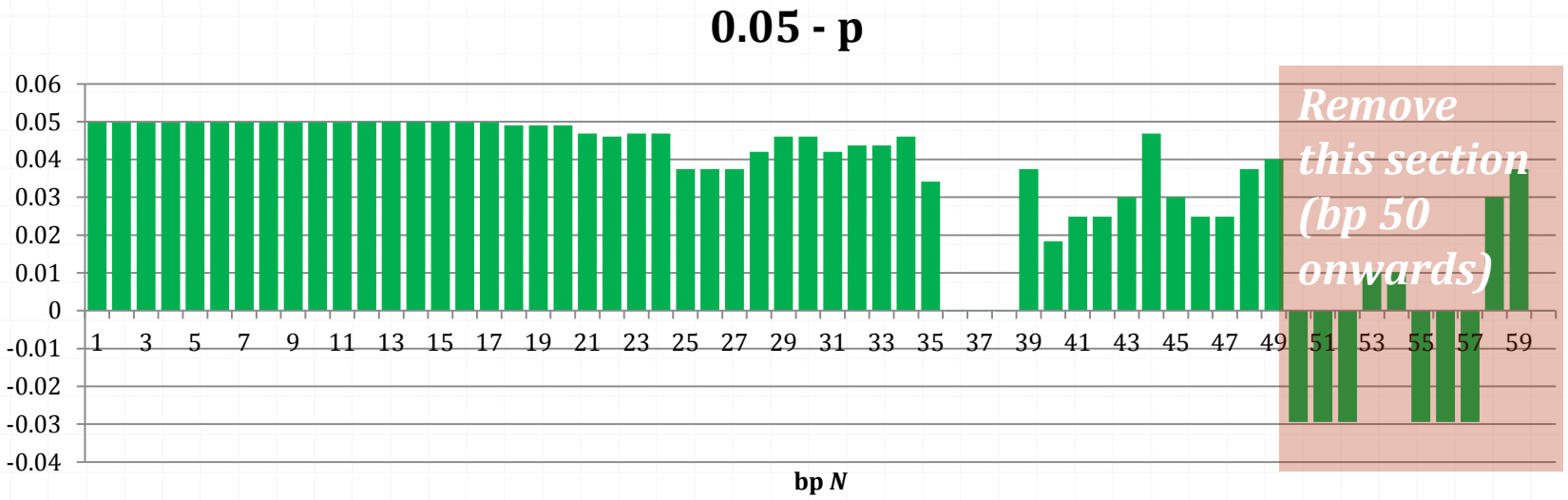


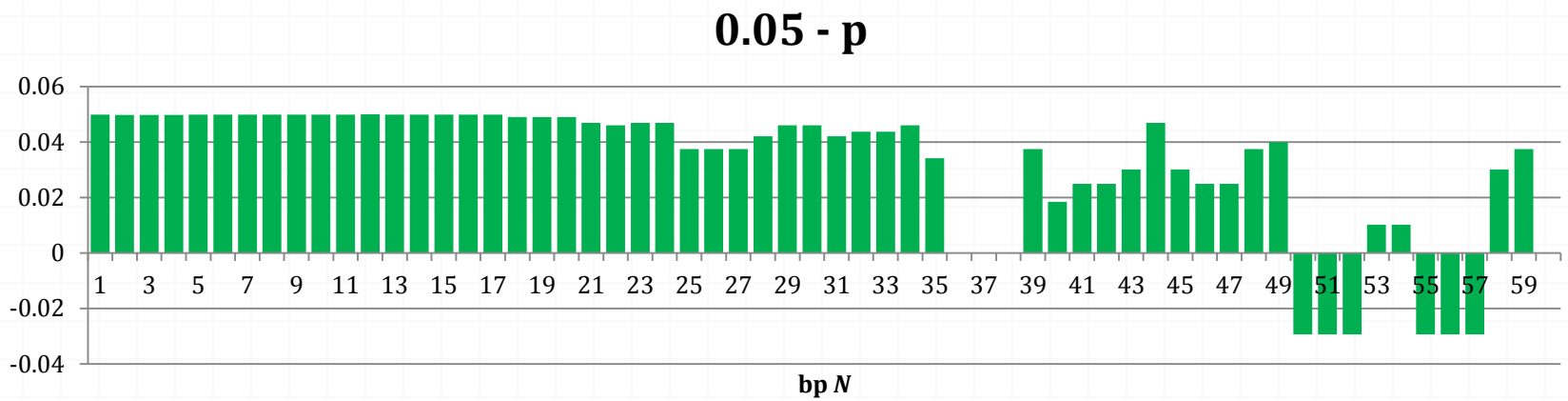
- Subtract the *p* values from a threshold value, e.g. 0.05



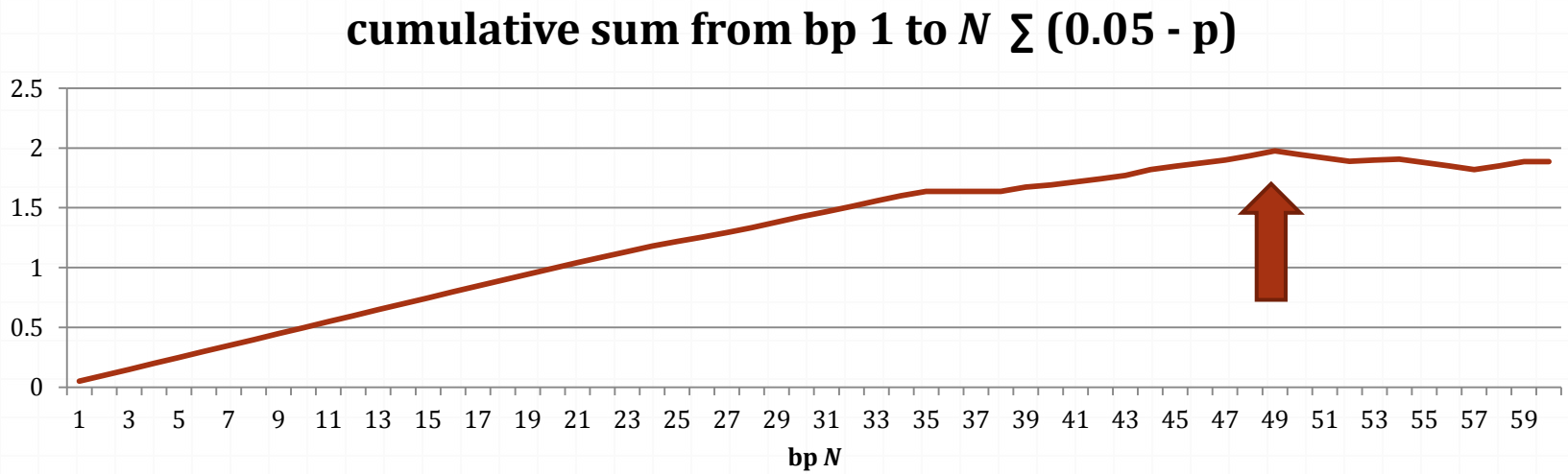


- Determine the subsequence with the **maximum sum**
- Remove everything else (i.e. to the left and right)





- For 3'-trimming only, this is equivalent to summing from left to right, and trimming after the maximum is reached



Example tools

- Some examples of window based:
  - PRINSEQ, PRINSEQ-LITE
  - FASTQ/FASTA trimmer of the FASTX* toolkit
  - Sickle
  - Trimmomatic
  - ConDeTri
  - FASTA+QUAL-format demultiplexer of Qiime
- Some examples of running sum-based:
  - phred program of PhredPhrap package
  - ERNE-FILTER
  - Cutadapt
- *not to be confused with the sequence similarity search program of the same name

Alternative rule-based methods

- E.g. consecutive bad scores
  - E.g. working from 5' to 3', trim (remove everything 3') when 3 consecutive Q values of < 20 occur
  - The qiime script for demultiplexing FASTQ format data sets uses this approach (c.f. the qiime script for demultiplexing FASTA+QUAL format data)