TEQC
An R-package for Quality Control in Target Capture Experiments

Manuela Hummel
manuela.hummel@crg.eu
Microarrays Unit, 4th floor, office 439.01
May 9th, 2011
Target Capture Experiments

- Sequencing complete genomes at high coverage is still expensive

- Targeted sequencing is cost-efficient approach for variant detection in genomic regions of interest

- Genomic DNA within regions of interest is captured (on microarrays or in solution) by pre-designed hybridization probes, and enriched previous to high-throughput sequencing

- Target regions can be commercial solutions (e.g. whole exome kits by Agilent, NimbleGen, Illumina) or customized genomic regions (e.g. linkage regions)
Target Capture Designs

• **Many small targets** (e.g. exons)
  -> hybridization probes (“baits”) will cover more or less all targeted bases

  vs

**Few large targets** (genomic regions)
  -> for repetitive regions usually no hybridization probes are designed

• **Large total target size** (e.g. whole exome ~50Mb)
  -> probe design without or little tiling

  vs

**Small total target size** (e.g. ~5Mb)
  -> probe design with higher tiling
Quality of Enrichment Process

• Besides standard sequencing quality control, the quality of the target enrichment process should be checked.

• Main quality issues in target capture experiments
  
  — **Capture Specificity**
  What fraction of sequenced reads fall on targeted regions?

  — **Capture Sensitivity**
  Do the targets have good coverage?

  — **Reproducibility**
  Are coverage distribution and coverage uniformity similar across replicates?

• The R/Bioconductor package **TEQC** was developed to address these and other issues.


---

**Reads**

<table>
<thead>
<tr>
<th>Reads</th>
<th>On-target</th>
<th>Off-target</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTCAGATCCACTAGGTT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGGTAAAGAAATTATAGG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTACTAACTAGGAGG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGAACTCTCGTCGGTCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGTGGCGCCCCCTTT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TGGAGCGTGAAAAGAAGT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCTACTGCGGGGAGGTTT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Reference genome**

```
GTCAGATCCACTAGGTT
CGGTAAAGAAATTATAGG
TTACTAACTAGGAGG
CGAACTCTCGTCGGTCT
CGTGGCGCCCCCTTT
TGGAGCGTGAAAAGAAGT
GCTACTGCGGGGAGGTTT
```

---

**Target**
Availability, Installation, Documentation

- **TEQC** is available from Bioconductor (from Release 2.8)
  

- and can be installed inside R by

  ```r
  > source("http://www.bioconductor.org/biocLite.R")
  > biocLite("TEQC")
  ```

- The package has to be loaded to the current session before usage

  ```r
  > library(TEQC)
  ```

- See the package “vignette” for complete documentation

  ```r
  > vignette("TEQC")
  ```

Or

http://www.bioconductor.org/packages/2.8/bioc/vignettes/TEQC.inst/doc/TEQC.pdf
Input Files

1. A file containing genomic positions of the target regions

```r
> targets <- get.targets(targetsfile="Targets.bed", 
  + chrcol=1, startcol=2, endcol=3, skip=0)
```

2. A file containing genomic positions of the aligned reads

```r
> reads <- get.reads("Reads.bed", chrcol=1, 
  + startcol=2, endcol=3, idcol=4, zerobased=F, skip=0)
```

3. (Optionally) a file containing genomic positions and sequences of the hybridization capture probes (baits)

```r
> baits <- get.baits("Baits.txt", chrcol=3, startcol=4, 
  + endcol=5, seqcol=2)
```
Note on Genomic Coordinates

- Bed files, as defined by UCSC (see http://genome.ucsc.edu/FAQ/FAQformat), follow the **0-based start / 1-based end** coordinate system
  - E.g. the first 100 bases on chromosome X have coordinates
    
    | chr  | start | end |
    |------|-------|-----|
    | chrX | 0     | 100 |

- BUT: bed files might also follow the **1-based** coordinate system!
  (e.g. results from Bowtie mapper)
  - The same region from above would have coordinates
    
    | chr  | start | end |
    |------|-------|-----|
    | chrX | 1     | 100 |

- **TEQC assumes** the “official” **0-based start / 1-based end** system and then **shifts** start coordinates **to the 1-based** system

- In case your files are already in 1-based system, set parameter **zerobased to FALSE** in order to avoid the shifting (as shown in previous example in get.reads())
Paired-End Data

- In case of paired-end data, some quality checks might rather be considered for read pairs instead of single reads.

- Reads can be “merged” by pairs.

```r
readpairs <- reads2pairs(reads)
```

- Reads without matching pair and pairs where both reads align to different chromosomes or too far apart from each other are not included.

- **Insert size** (= length from start of first read to end of second read) **distribution**

```r
insert.size.hist(readpairs)
```
Specificity

• Is the number of reads mapping to each chromosome proportional to the amount of targeted bases?

> chrom.barplot(reads, targets)

• Fraction of aligned reads that overlap target regions (by at least 1 base)

> fraction.reads.target(reads, targets)

• Consider also e.g. 50 bases on both sides of each target in the calculation

> fraction.reads.target(reads, targets, Offset=50)

• For paired-end data, fraction of read pairs that overlap target regions (i.e. where at least one of the two reads overlaps a target)

> fraction.reads.target(readpaires, targets)
Enrichment

• Measure for the **enrichment of sequences within the targeted** region

• Depends on fraction of reads mapping to the target and total target size relative to the respective genome size

\[
\frac{\text{# reads on target}}{\text{# aligned reads}} \times \frac{\text{target size}}{\text{genome size}}
\]

```
> fr <- fraction.reads.target(reads, targets)
> ft <- fraction.target(targets, genome="hg19")
> enrichment <- fr / ft
```

• In case of paired-end data, enrichment can also be calculated for read pairs instead for single reads
Coverage

> Coverage <- coverage.target(reads, targets)

• **Overall on-target** coverage average, standard deviation, quantiles
  > Coverage$avgTargetCoverage
  > Coverage$targetCoverageSD
  > Coverage$targetCoverageQuantiles

• **Coverage average and standard deviation per target**
  > Coverage$targetCoverages

• **Coverage per targeted base**
  > Coverage$coverageTarget

• **Coverage per sequenced and/or targeted base**
  > Coverage$coverageAll

• **Number of reads** overlapping each **target**
  > readsPerTarget(reads, targets)

• All coverage calculations are based on single reads!
Sensitivity

• What fraction of target bases is covered by at least 1, 2, 5, 10, ... reads?

> covered.k(Coverage$coverageTarget)

• Coverage histogram: graphical presentation of per-target-base coverage

> coverage.hist(Coverage$coverageTarget, + covthreshold=8)

~75% of targeted bases have at least 8X coverage
Coverage Uniformity

- Is the coverage uniform across targeted bases?
- Figure is based on **normalized coverage**

\[
\text{Per-base coverage} = \frac{\text{Average coverage over all target bases}}{\text{Normalized coverage}}
\]

- Normalized coverage is not dependent on absolute quantity of sequenced reads and therefore better comparable between samples / experiments

\[\text{coverage.uniformity(Coverage)}\]

Fraction of targeted bases that reach at least
- **half the average** normalized coverage (=0.5)
- **average** normalized coverage (=1)

The steeper the curve is falling, the less uniform is the coverage.
Coverage vs Target Length

• Is the **number of reads** mapping to a target proportional to its size?

-> Expected, since for larger targets there should be more capture probes

```r
> RpT <- readsPerTarget(reads, targets)  
> coverage.targetlength.plot(RpT,  
  + plotcolumn="nReads", pch=16, cex=1.5)
```

• Does the **average coverage** depend on target size?

-> E.g. small targets might have smaller coverage due to worse bait tiling as compared to larger targets

```r
> avgC <- Coverage$targetCoverages  
> coverage.targetlength.plot(avgC,  
  + plotcolumn="avgCoverage", pch=16, cex=1.5)
```
Coverage vs GC Content

- Does **coverage** depend on **GC content** of the capture probes

  → Expected, since capture probes with very high or very low GC content have worse hybridization properties

  ```r
  > baits <- get.baits("Baits.txt", chrcol=3, + startcol=4, endcol=5, seqcol=2)
  > coverage.GC(Coverage$coverageAll, baits, pch=16, + cex=1.5)
  ```

Average (normalized) coverage **per bait**
Coverage along Genome

- **Display per-base coverage** along a genomic interval
- Highlight target regions

```r
> coverage.plot(Coverage$coverageAll, targets, + Offset=100, chr="chr1", Start=11157524, + End=11158764)
```

- ... or export per-base coverage to **wiggle files** for use of genome browsers

```r
> make.wigfiles(Coverage$coverageAll)
```
Reproducibility

• Target coverage of (technical) replicates should be similar

\[
> \text{coverage.correlation(covlist, } \\
> \quad + \text{ plotfrac=0.1, cex.pch=4)}
\]

• Coverage densities

\[
> \text{covlist } \leftarrow \text{list(Coverage, Coverage2)} \\
> \text{coverage.density(covlist, normalized=F)}
\]

• Coverage uniformity

\[
> \text{coverage.uniformity(Coverage, } \\
> \quad + \text{ addlines=F)} \\
> \text{coverage.uniformity(Coverage2, } \\
> \quad + \text{ addlines=F, add=T, col="blue", lty=2)}
\]

• Coverage along genomic region

\[
> \text{coverage.plot(Coverage$coverageAll, } \\
> \quad + \text{ add=T, col.line=2, ...}
\]
Duplicate Analysis

• Read duplicates, i.e. reads with same start and end positions, are usually **removed** before follow-up analysis because they are supposed to be **PCR artefacts**

• Problematic for Target Capture experiments, because many **“real” duplicates**, i.e. reads derived from actually different DNA molecules, are expected due to the **enrichment** process!

> duplicates.barplot(reads, targets)

>20% of **on-target** reads are present in 2 copies
almost 10% in 3 copies

in **off-target** reads a much lower fraction is duplicated

• After removing duplicates the per-base **coverage is limited** by the read length!

E.g. if the read length is 30, a base can only be covered by **at most** 30 reads starting at **different** positions
Duplicate Analysis

“Solutions”

• Remove duplicates only if also read **sequences** are identical (outside of TEQC)

• Use **long reads**

• Use **paired-end** sequencing
  → a read pair is only considered a duplicate if start and end positions of *both* reads are identical to those of another read pair
  → less likely to occur for different DNA fragments

```r
> duplicates.barplot(readpairs, + targets, + ylab="Fraction of read pairs")
```
Future Work

• Provide automated HTML or PDF report that can be evoked by just one command line

• Allow BAM file format for the input files

• Improvements in terms of required computing time and memory

• Further functionalities, e.g. functions to retrieve easily average / per-base coverage for a selected region
References


### Acknowledgements

**Microarrays Unit**
- Sarah Bonnin
- Anna Ferrer
- Heidi Mattlin
- Mònica Bayés

**Bioinformatics Core**
- Guglielmo Roma
- Ernesto Lowy

**Ultrasequencing Unit**
- Heinz Himmelbauer
- Debayan Datta
- Matthew Ingham

**Genes and Disease**
- Xavier Estivill
- Eva Riveira
- Raquel Rabionet
- Daniel Trujillano