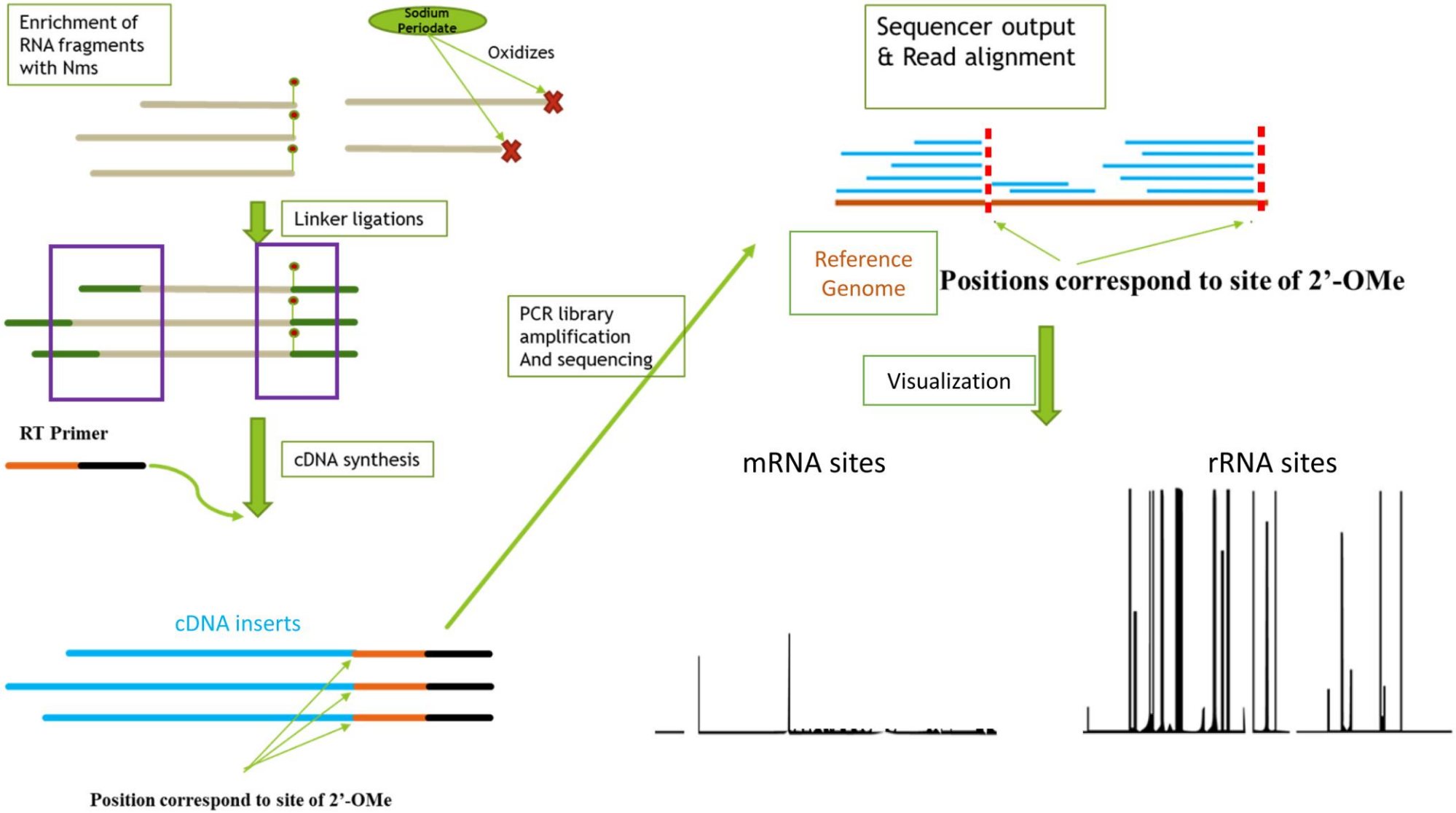
Demonstration of basic RibOxi-seq concept



RNA structure after 3’ and 5’ linker ligations

5'-/Biosg/ACACUCUUUCCCUACACGACGCUCUUCCGAUCUNNNN------insert-------ATCACGCTGTAGGCACCATCAATGACAG/SpC3/- 3´

cDNA library structure

Randomer for deduplication

3’-linker sequence

3'-TGTGAGAAAGGGATGTGCTGCGAGAAGGCTAGANNNN----insert----TAGTGCGACATCCGTGGTAGTTACTGTCTAGCCTTCTCGTGTGCAGACTTGAGGTCAGTG-5'

Complementary to partial Illumina i5 PCR primer

In-line barcode for filtering out RT mis-priming

Identical to partial Illumina i7 PCR primer

dsDNA library structure

Forward	**5'-Illumina_i5_adapter**NNNN----insert----**ATCACGCTGTAGGCACCATCAATGACIllumina_i7_adapter-3'**
Reverse	**3'-Illumina_i5_adapter**NNNN----insert----**TAGTGCGACATCCGTGGTAGTTACTGIllumina_i7_adapter-5'**

Sequencing Output (example using 1*150 single-end sequencing)

Read:

```
@ Sequence_identifier+i7_index+i5_index
NNNN----INSERT----ATCACGCTGTAGGCACCATCAATGAC---MAYBE_READ-THROUGH_INTO_ILLUMINA_PCR_ADAPTER
+
_QUALITY_OF_THE_READ_____________________________________________________________________
```

1. *cutadapt* used to remove partial linker sequence and read-through adapters, discarding reads without match:

Read output:

```
@ Sequence_identifier+i7_index+i5_index
NNNN----INSERT----ATCACG
+
_QUALITY_OF_THE_READ____
```

2. *move_umi.py* moves randomer sequence from read to read-identifier:

```
@ Sequence_identifier_NNNN+i7_index+i5_index
INSERT----ATCACG
+
_OF_THE_READ____
```

3. *cutadapt* used to remove the barcode portion of linker and discard any read without a perfect match:

```
@ Sequence_identifier_NNNN+i7_index+i5_index
INSERT
+
_OF_TH
```

4. *STAR* alignment to genome of interest

5. *samtools* indexing the bam output

6. umi_tools deduplication:

Example aligned read #1:
UMI: AATC
Sequence: GGTTACG

Example aligned read #2:
UMI:CGTA
Sequence: GGTTACG

Not duplicates

Example aligned read #1:
UMI: AATC
Sequence: GGTTACG

Example aligned read #2:
UMI: AATC
Sequence: GGTTACG

Duplicates, keep only 1 copy

7. *bedtools* converts deduplicated bam to bed file format

8. *genomecov* generates genome coverage track of 3'-end only to visualize Nm sites on genome browser

9. *riboxi_bed_parsing.py* counts all 3’-end alignments and generates a tab delimited file listing chromosome number, base position, gene name and counts from the bed file:

chr	base	gene	WT5_R	DKO3	DKO3_R	WT5
chr1	13236	DDX11L1	0	1	0	0
chr1	13238	DDX11L1	0	0	0	1
chr1	14405	WASH7P	0	5	0	1
chr1	14406	WASH7P	0	3	0	0
chr1	14408	WASH7P	0	1	0	1
chr1	14409	WASH7P	0	1	0	0
chr1	14413	WASH7P	0	1	0	1

10. The shinyapp allows more interactive interrogation of data such as filtering and visualization on the fly: