

Steps to run RSW:

```
#Pre-proces: Parse UCSC refFlat.txt file
perl refflat_parse_RSW.pl refFlat.txt
```

```
## 1) Align original reads
```

```
bowtie BOWTIE_INDEX_FILE --solexa1.3-quals -p 14 -v 2 -k 11 -m 10 --best -q Het.txt --un
Het.unmapped.txt --max Het.max.txt Het.bowtie.txt
bowtie BOWTIE_INDEX_FILE --solexa1.3-quals -p 14 -v 2 -k 11 -m 10 --best -q KO.txt --un
KO.unmapped.txt --max KO.max.txt KO.bowtie.txt
```

```
## 2) Split unmapped reads (At the same time, the script trims the last 2bp for each
unmapped read)
```

```
perl split_read_RSW.pl Het.unmapped.txt 22 11
perl split_read_RSW.pl Het.unmapped.txt 21 12
perl split_read_RSW.pl Het.unmapped.txt 20 13
perl split_read_RSW.pl Het.unmapped.txt 19 14
perl split_read_RSW.pl Het.unmapped.txt 18 15
perl split_read_RSW.pl Het.unmapped.txt 17 16
perl split_read_RSW.pl Het.unmapped.txt 16 17
perl split_read_RSW.pl Het.unmapped.txt 15 18
perl split_read_RSW.pl Het.unmapped.txt 14 19
perl split_read_RSW.pl Het.unmapped.txt 13 20
perl split_read_RSW.pl Het.unmapped.txt 12 21
perl split_read_RSW.pl Het.unmapped.txt 11 22
cat Het.unmapped.txt.split.* >Het.unmapped.txt.cut2.split
perl split_read_RSW.pl KO.unmapped.txt 22 11
perl split_read_RSW.pl KO.unmapped.txt 21 12
perl split_read_RSW.pl KO.unmapped.txt 20 13
perl split_read_RSW.pl KO.unmapped.txt 19 14
perl split_read_RSW.pl KO.unmapped.txt 18 15
perl split_read_RSW.pl KO.unmapped.txt 17 16
perl split_read_RSW.pl KO.unmapped.txt 16 17
perl split_read_RSW.pl KO.unmapped.txt 15 18
perl split_read_RSW.pl KO.unmapped.txt 14 19
perl split_read_RSW.pl KO.unmapped.txt 13 20
perl split_read_RSW.pl KO.unmapped.txt 12 21
perl split_read_RSW.pl KO.unmapped.txt 11 22
cat KO.unmapped.txt.split.* >KO.unmapped.txt.cut2.split
```

3) Align split unmapped reads

```
bowtie BOWTIE_INDEX_FILE --solexa1.3-quals -p 14 -v 0 -k 11 -m 10 --best -q
Het.unmapped.txt.cut2.split --un Het.unmapped.txt.cut2.split.unmapped.txt --max
Het.unmapped.txt.cut2.split.max.txt Het.unmapped.txt.cut2.split.bowtie.txt
bowtie BOWTIE_INDEX_FILE --solexa1.3-quals -p 14 -v 0 -k 11 -m 10 --best -q
KO.unmapped.txt.cut2.split --un KO.unmapped.txt.cut2.split.unmapped.txt --max
KO.unmapped.txt.cut2.split.max.txt KO.unmapped.txt.cut2.split.bowtie.txt
```

4) Break down the first column into readName, readDirection, readLength...

```
perl split_1stColumn_RSW.pl Het.unmapped.txt.cut2.split.bowtie.txt
perl split_1stColumn_RSW.pl KO.unmapped.txt.cut2.split.bowtie.txt
```

5) select candidate split pairs

```
SELECT s1.readname, s1.splitend, s1.splitsize, s1.strand, s1.chromosome,
s1.position, s1.nummatches, s2.readname, s2.splitend, s2.splitsize, s2.strand,
s2.chromosome, s2.position, s2.nummatches, ABS(s1.position - s2.position) "distance"
FROM DB$DWarehouse.split_read_alignr46l6_ntrnrmc2 s1
JOIN DB$DWarehouse.split_read_alignr46l6_ntrnrmc2 s2 ON s1.strand = s2.strand
and s1.chromosome = s2.chromosome WHERE s1.splitend = 'L' AND ABS(s1.position -
s2.position) < 40000
and s1.readname = s2.readname AND s1.splitend != s2.splitend AND (s1.splitsize + s2.splitsize)
= 33;
```

```
SELECT s1.readname, s1.splitend, s1.splitsize, s1.strand, s1.chromosome,
s1.position, s1.nummatches, s2.readname, s2.splitend, s2.splitsize, s2.strand,
s2.chromosome, s2.position, s2.nummatches, ABS(s1.position - s2.position) "distance"
FROM DB$DWarehouse.split_read_alignr46l7_ntrnrmc2 s1
JOIN DB$DWarehouse.split_read_alignr46l7_ntrnrmc2 s2 ON s1.strand = s2.strand
and s1.chromosome = s2.chromosome WHERE s1.splitend = 'L' AND ABS(s1.position -
s2.position) < 40000
and s1.readname = s2.readname AND s1.splitend != s2.splitend AND (s1.splitsize + s2.splitsize)
= 33;
```

Or alternatively, users can use "find_pair_RSW.pl" to run this step instead of using SQL commands.

```
perl find_pair_RSW.pl Het.unmapped.txt.cut2.split.bowtie.txt.split1stcolumn 40000 33
perl find_pair_RSW.pl KO.unmapped.txt.cut2.split.bowtie.txt.split1stcolumn 40000 33
```

Notes: Output file names: Het.unmapped.txt.cut2.split.bowtie.txt.split1stcolumn.txt AND KO.unmapped.txt.cut2.split.bowtie.txt.split1stcolumn.txt
(Demo files: we have Het.unmapped.txt.cut2.split.bowtie.txt.split1stcolumn.chr11.txt AND KO.unmapped.txt.cut2.split.bowtie.txt.split1stcolumn.chr11.txt)

6) Filter out adjacent pairs and select pairs falling within the same UCSC gene at the same time

```
perl check_pair_RSW.pl Het.unmapped.txt.cut2.split.bowtie.txt.split1stcolumn.chr11.txt
refFlat.txt
perl check_pair_RSW.pl KO.unmapped.txt.cut2.split.bowtie.txt.split1stcolumn.chr11.txt
refFlat.txt
```

7) Consolidate reads

- I) Filter out regions (< 3 splice_length) and their support reads AND add spliced region information;
- II) Cluster reads;
- III) Check/map against KnownGene annotation file or classify into non-canonical or novel spliced regions;
- IV) Compare both samples side-by-side and pull out read sequence information for identified spliced regions

```
perl make_table_RSW.pl
Het.unmapped.txt.cut2.split.bowtie.txt.split1stcolumn.chr11.txt.results
KO.unmapped.txt.cut2.split.bowtie.txt.split1stcolumn.chr11.txt.results Het.txt KO.txt
refFlat.txt.intronBoundary.exonsgaps 11 2 5
```

8) Post-processing step (filter out identified regions with 2 or more reads support)

```
perl filter_2orMoreReads.pl Output.duplicateRemoval.txt 2
OR
awk '$3>=2{print $1, "\t", $2, "\t", $3, "\t", $4, "\t", $5, "\t", $6, "\t", $7, "\t", $8, "\t", $9, "\t",
$10}' Output.duplicateRemoval.txt >Output.duplicateRemoval.2orMoreReads.txt
```

The final output file is Output.duplicateRemoval.2orMoreReads.txt.