PeakRescue pipeline

Run PeakRescue as follows:

cd peakRescue/

```
perl scripts/perl/bin/runPeakRescue.pl \
-bam datasets/chr21.bam \
-gtf datasets/chr21.gtf.gz \
-g datasets/chr21.fa \
-alg clipover \
-o results_dir
```

Description of the steps within the pipeline [User can see these in peakrescue.log file]

A. **PeakRescue: extract unique reads using HTSeq and disambiguate/rescue ambiguously mapped reads.**

HTSeq package included in Peakrescue https://github.com/rnaseq/peakrescue, contains the original version of count.py (as of HTSeq version ***) and the extended version of count.py (*i.e.* count_peakRescue.py) which implements the disambiguation and rescue steps of peakrescue pipeline.

The input BAM file must be produced with a splice-aware aligner and must contain "NH:i:" tags (see SAM format specifications) .

1. **Extract the non-ambiguous uniquely mapped reads**

Run HTSeq on the input BAM file with the following mandatory options: --*samout* as this permits to generate an output SAM file that is used in the next step of the pipeline; and the two additional options: --*type*=exon and --*idattr*=gene_id, to specify that peakRescue performs read summarization and rescue of ambiguous reads at the gene level.

Run as follows:

```
samtools sort –on accepted_hits.bam tmpsort | samtools view - | \
python \
count.py \
--mode=union \
--stranded=no \
--samout= htseq.sam \
--type=exon \
--idattr=gene_id \
- \
genome.gff \
>htseq_count.out
```

Output files:

hits_htseq.sam :

SAM output file containing reads with an added *XF:Z* tag showing status of a given read in one of the three categories:
XF:Z:<gene_name> - read assigned to a single gene.
XF:Z:ambiguous[<gene_name_1>...<gene_name_n>] – Uniquely mapped ambiguous read overlapping more than one gene.
XF:Z:alignment_not_unique – multi mapped reads mapping at more than one location.

htseq_count.out: Tab separated file containing per gene fragment count data for unique fragments.

2. **Disambiguate ambiguous unique reads and store the remaining ambiguously mapped reads (unique and multimapped)**

The reads flagged with either " *XF:Z:ambiguous*" or "*XF:Z:alignment_not_unique*" in the SAM file (output generated in the previous step –see A.1 section) are used as input to the extended version of the HTSeq count.py script (count_peakRescue.py).

Run as follows:

```
grep -P "ambiguous|alignment_not_unique" htseq.sam | \
python \
count_peakRescue.py \
--mode=union \
--stranded=no \
--samout=disambiguated.sam \
--type=exon \
--idattr=gene_id \
- \
genome.gff \
multimapped_readname_gene_name.out \
ambiguous_readname_gene_name.out \
> disambiguated_count.out
```

Output files:
disambiguated.sam: SAM file containing unique disambiguated reads assigned to a single gene labelled in the *XF:Z:<gene_name>* tag.
multimapped_readname_gene_name.out: This is an intermediate tab separated file, which provides read names-to-gene names mappings – with two columns:
Column 1: Read name
Column 2: A list of gene names (Ensembl IDs) on which the given multimapped read maps.
ambiguous_readname_gene_name.out: This is an intermediate tab separated file, which provides read names-to-gene names mappings – with two columns:
Column 1: Read name
Column 2: A list of gene names (Ensembl IDs) on which the given ambiguous read overlaps.
rescued_count.out: Tab separated file containing per gene fragment count data with the following columns:
Column 1: Ensembl ID
Column 2: Unique disambiguated,

Column 3: Ambiguous [multimapped + ambiguous unique]
Column 4: The total read count for the given gene. Fraction of non-unique read will be assigned to underlying genes listed in *multimapped_readname_gene_name.out* and *ambiguous_readname_gene_name.out* files based on expression proxy defined by maximum unique peak[*Gene expression proxy* in Methods].

B. **Pre-process GTF file:**
Run as follows:
processGTF.pl -gtf genome.gtf.gz

The input GTF file may be downloaded from Ensembl and should contain transcripts and gene information for a given species.
Process the input GTF file in order to get the "global transcript" for each gene described in the input GTF (*i.e.* merged exons over all transcripts of a gene). [see Fig.1 A – paper under submission]. This GTF pre-processing steps creates the following files to be used in the later steps of PeakRescue:

1. *global_transcript.bed*: Exon interval for a single transcript created after merging all known transcripts for a gene.
2. *unique_regions.bed*: unique and non overlapping intervals per gene.
3. *geneboundaries.bed*: start and stop of a global transcript defined for respective genes.
4. *global_transcript_gene_length.tab:* gene length based on all non overlapping exons listed in global_transcript.bed file.
5. *unique_segment_gene_length.tab*: gene length based on unique regions listed in unique_regions.bed file.

C. **Calculate peak [see methods]**
Input BAM file is a merged BAM containing the non-ambiguous uniquely mapped and the disambiguated uniquely mapped reads.
getPeak.pl -bed geneboundaries.bed -bam merged.bam -g genome.fa

Output file:
peak.tab: Tab separated file containing gene name and peak value

D. **Run probabilistic assignment of ambiguous reads to underlying genes.**
Run as follows:
*python peakRescue_readToGeneAssignment.py *
*-p peak.tab *
*-m READTYPE_readname_gene_name.out *
*-l global_transcript_gene_length.tab *
-t READTYPE

Output file:
results_peakrescue_readtype_[READTYPE]_all_genes.out is produced with the tag RT replaced by the read type *e.g.* "ambiguous_unique" or "multimapped" used in the command line (-t option).
Tab separated output file contains the following columns:
Column 1: Ensembl ID
Column 2: Proportion of [readType] assigned to gene

E. **Generate peakRescue output file.**

The peakRescue output file is generated using the mergeFile.pl script to combine the HTSeq read counts, the disambiguated read counts and the proportions of ambiguous unique and multimappers rescued.

peakRescueFinalCount.out : This file contains per gene count values that user can input to any differential expression analysis algorithms. The *totalCount* column contains sum of unique*,* unique disambiguated and peakRescue contributions (*i.e.* sum of all rescued ambiguous reads' contributions).

The output file, *peakRescueFinalCount.out*, contains the following columns:

Gene: gene name as specified in the original input gtf file
uniqueCount: Unique read count
uniqueDisambiguatedCount: Disambiguated unique read count
ambiguousUniqueToRescue: Ambiguous unique reads to rescue
ambiguousUniqueProportion: Proportion of ambiguous unique reads rescued
multimapperToRescue: Multimapper reads to rescue
multimapperProportion: Proportion of multimapper reads rescued
totalCount: Total read count *i.e.* sum of *uniqueCount, uniqueDisambiguatedCount, ambiguousUniqueProportion and multimapperProportion.*
GlobalTranscriptLength: Gene length based on the global transcript
fpkmTotalCount: totalCount converted to FPKM using the *GlobalTranscriptLength.*