新RNA-seq基因组数据准备

- 1 gene_info
  - transcript.fa（通过gffread从gtf和fa文件提取）
  - gene.txt（首先通过featureCounts获得feature_count.txt；然后提取第1，6列信息得到gene_length.txt，gene中的长度是每个exon的长度和；最后根据物种是否在dataset数据集中，在的话gene_description需要从biomaRt中提取（需要联网），不在就直接从gtf文件提取）
  - transcript.txt（方法同gene.txt相似，只是不需要gene_length.txt文件，添加transcript_description和gene.txt方法一致）
  - flat.txt（通过gtfToGenePred从gtf文件中提取，首先得到第一列为transcript_id和各exon和CDS的位置信息文件，然后根据transcript.txt将gene_id添加至第一列）
  - gene2pep.txt（根据gtf文件提取gene_id、transcript_id和protein_id三列信息）
  - gene.fa（根据transcript.fa和transcript.txt两个文件提取每个gene的序列信息，取每个gene_id中最长转录本的序列作为基因序列）
- 2 index_and_gzip
  - fa.fai（samtools faidx *.fa命令建立一个后缀为.fai 的文件，根据这个.fai 文件和原始的fa文件， 能够快速的提取任意区域的序列）
  - dict（picard对fa生成dict文件）
  - ht2（hisat2-build建立fa的hisat2索引）
  - gtf.gz、fa.gz、transcript.fa.gz、gene.fa.gz（压缩文件）
- 3 gene_go
  - ensemble：首先通过biomaRt包联网至biomart数据库，然后从中提取该物种数据集的ensembl_gene_id和go_id两列信息，得到go.txt文件；然后从GO.db中对go.txt添加go_ontology和go_term两列信息，得到最终的go.txt文件
  - ncbi：物种taxon在ncbi2go.txt中（根据taxon编号直接从准备好的gene2go.txt文件提取前三列信息，然后从GO.db中对go.txt添加go_ontology和go_term两列信息，得到最终的go.txt文件）
  - other：其他数据库、物种taxon不在ncbi2go.txt中（首先使用TransDecoder软件对gene.fa识别编码区并预测蛋白：识别长度至少为100个氨基酸的开放阅读框，后通过blast对比swissprot蛋白数据库和Pfam搜索已知蛋白的同源序列来识别ORF，后预测可能的编码区；然后使用interproscan将蛋白序列与数据库进行比对注释得到go.txt；最后从GO.db中对go.txt添加go_ontology和go_term两列信息，得到最终的go.txt文件）
- 4 gene_kegg
  - ensemble：首先通过biomaRt包联网至biomart数据库，然后从中提取该物种数据集的gene_id、pathway_gene_id、pathway_id和pathway_name四列信息，得到kegg.txt；最后下载物种的通路文件（html、png、xml）
  - ncbi：使用KEGGREST包下载kegg_gene_id、kegg_pathway_id和kegg_pathway_name，对于ncbi数据中gene_id是数字格式的，gene_id和kegg_gene_id一致的，所以最后生成的kegg.txt前两列信息是一样的；最后下载物种的通路文件（html、png、xml）
  - kobas：对于不属于ensemble和ncbi数据库中的物种，如果物种在kobas数据库中，进行kobas的注释。首先通过blast的方法对gene.fa与kobas数据库中的物种序列进行比对得到abbr_annotate.txt（两列信息，gtf文件的gene_id和kobas数据库中的信息Gene ID|Gene name|Hyperlink），然后使用KEGGREST包获取kegg_gene_id、pathway_id和pathway_name，将kegg_gene_id与gene_id进行关联，获得kegg.txt；最后下载物种的通路文件（html、png、xml）
  - other：由于kegg数据库中物种比kobas要多，当物种不在kobas物种编号中时（比如烟草），进行如下注释：首先使用KEGGREST包获取物种的核酸序列，然后使用makeblastdb对核酸序列进行格式化数据库，之后将gene.fa与核酸序列数据库进行一个blastn比对，得到blast.txt，后续与kobas过程一致
- 5 gene_ppi
  - ensemble：提取gene2pep.txt中的gene_id和protein_id，并在protein_id前加入taxon编号（如9606.），得到ppi.txt
  - ncbi：物种taxon在ncbi2ens.txt中，直接从配置文件gene2ensembl.txt中提取第一列和第七列的信息，并在protein_id前加入taxon编号（如9606.），得到ppi.txt
  - other：使用diamond将gene.fa与string数据库中的taxon__protein.sequences.dmnd进行blastx比对，得到gene_id与数据库中protein_id的对应关系，最后提取出与protein_id比对evalue最小的gene_id，得到ppi.txt
- 6 gene_tf
  - animal（latin在animaltf.txt中）
    - ensemble：从配置文件latin_tf.txt提取第一列和第三列得到tf.txt
    - ncbi：从配置文件latin_tf.txt提取第二列（非\N）和第三列得到tf.txt
    - other：使用生成go.txt中的中间文件pep.tsv，筛选pep.tsv中第五列在配置文件tf_anno.txt第一列的PfamID，获得PfamID与tf_anno.txt第二列tf_Family的对应关系，同时找到每个gene_id对应的evalue最小的PfamID，最终得到gene_id与tf_Family的对应关系生成tf.txt
  - plant（latin在planttf.txt中）
    - ensemble/ncbi：从配置文件latin_tf.txt提取第二列和第三列得到tf.txt
    - other：和动物一样，使用生成go.txt中的中间文件pep.tsv，筛选pep.tsv中第五列在配置文件tf_anno.txt第一列的PfamID，获得PfamID与tf_anno.txt第二列tf_Family的对应关系，同时找到每个gene_id对应的evalue最小的PfamID，最终得到gene_id与tf_Family的对应关系生成tf.txt
  - other：如果生成go.txt中使用的是interproscan进行的注释，则准备tf时直接使用中间文件pep.tsv；如果不是则需要先生成pep.tsv，后续过程与前面一致
- 7 check
  - 基因组fa：不能有空行
  - 基因组gtf：第一列染色体名在fa文件中；第四列和第五列的值不能大于fa文件对应染色体的长度；gene_id和transcript_id不能一样
  - gene.fa、gene.xls：gene.xls文件列数是10列；gene.fa中的序列id的数目要与gene.xls第一列gene_id的数目一样
  - transcript.fa、transcript.txt：transcript.txt文件列数10列；transcript.fa中的序列id的数目要与transcript.txt第一列transcript_id的数目一样
  - go.txt：4列文件
  - kegg.txt：4列文件；第三列文件的xml在/NJPROJ1/RNA/database/kegg/abbr/下存在；文件第一列的id在gene.xls中的数目要大于100
  - ppi.txt：文件第二列中的protein_id在/NJPROJ1/RNA/database/string/abbr/abbr__proteins.tsv.gz的数目需要大于100
  - tf.txt：文件第一列id不在gene.xls中gene_id列表中的数目要小于10