Apr 05, 2022

RatGTEx pipeline

DOI
https://dx.doi.org/10.17504/protocols.io.rm7vzyk92lx1/v1

Daniel Munro[1], Laura M Saba[2], Hao Chen[3], Abraham Palmer[1], Pejman Mohammadi[4]

[1]University of California, San Diego; [2]University of Colorado Anschutz Medical Campus; [3]University of Tennessee Health Science Center; [4]Scripps Research

CGORD

Daniel Munro
University of California, San Diego, Scripps Research

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN ACCESS

DOI: https://dx.doi.org/10.17504/protocols.io.rm7vzyk92lx1/v1

External link: https://ratgtex.org/

Protocol Citation: Daniel Munro, Laura M Saba, Hao Chen, Abraham Palmer, Pejman Mohammadi 2022. RatGTEx pipeline. **protocols.io** https://dx.doi.org/10.17504/protocols.io.rm7vzyk92lx1/v1

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working
We use this protocol and it's working

Created: April 04, 2022

Last Modified: April 05, 2022

Protocol Integer ID: 60302

Keywords: rat, eQTL, gene expression, RNA-Seq, rodent, outbred, ratgtex pipeline, data for the ratgtex portal, code for this pipeline, gtex eqtl mapping pipeline, ratgtex portal, output file name, pipeline, utility script, python, framework for reproducible data analysis, process, command, reproducible data analysis, data

Funders Acknowledgements:
National Institute on Drug Abuse
Grant ID: P50DA037844
National Institute on Drug Abuse
Grant ID: P30DA044223
National Institute of General Medical Sciences
Grant ID: R01GM140287
National Institute on Alcohol Abuse and Alcoholism
Grant ID: R24AA013162

Abstract

This is the pipeline used to process data for the **RatGTEx Portal**. It is loosely based on the **GTEx eQTL mapping pipeline**, though it includes some utility scripts from there in their entirety. All code for this pipeline can be found in the **repository**. It is built on **Snakemake**, a Python-based framework for reproducible data analysis. The commands reproduced here use Snakemake-style templating, with variables in brackets to represent different input/output file names and parameters.

Align RNA-Seq reads

1 Generate the index for STAR.

Software

STAR NAME

https://github.com/alexdobin/STAR SOURCE LINK

Command

STAR index

```
STAR --runMode genomeGenerate \
     --genomeDir {params.outdir} \
     --genomeFastaFiles {input.fasta} \
     --sjdbGTFfile {input.gtf} \
     --sjdbOverhang {params.overhang} \
     --runThreadN {resources.cpus}
```

2 Get an individual-specific VCF file. This is used by STAR to consider the individual's variants for better alignment.

Software

bcftools

NAME

https://github.com/samtools/bcftools

REPOSITORY

https://github.com/samtools/bcftools

SOURCE LINK

Command

bcftools view

```
bcftools view -s {wildcards.rat_id} --min-ac=1 -O z -o {output}
{input}
```

3 Align RNA-Seq reads for a sample using STAR.

Software

STAR

NAME

https://github.com/alexdobin/STAR

SOURCE LINK

Command

STAR align

```
STAR --runMode alignReads \
        --runThreadN {resources.cpus} \
        --genomeDir {params.index_dir} \
        --readFilesIn {params.fastq_list} \
        --readFilesCommand zcat \
        --twopassMode Basic \
        --varVCFfile <(zcat {input.vcf}) \
        --waspOutputMode SAMtag \
        --quantMode TranscriptomeSAM \
        --outSAMstrandField intronMotif \
        --outSAMattrRGline {params.read_groups} \
        --outSAMtype BAM SortedByCoordinate \
        --outSAMunmapped Within \
        --outFileNamePrefix {params.prefix}
```

Identify and correct sample mixups

4 Get all exon regions from gene annotations.

Command

Exons from GTF annotations

```
grep -v '^#' {input} | awk '$3=="exon"' | cut -f1,4,5 | gzip -c >
{output}
```

5 Subset genotypes to only exon SNPs, SNPs with MAF >= 0.2, and SNPs with <10% missing values.

Software

bcftools

NAME

https://github.com/samtools/bcftools

REPOSITORY

https://github.com/samtools/bcftools

SOURCE LINK

Command

bcftools view

```
bcftools view {input.vcf} \
    --regions-file {input.regions} \
    -Ou | bcftools view \
    --min-af 0.2:minor \
    -i 'F_MISSING<0.1' \
    -Oz -o {output.vcf}
```

Software

samtools

NAME

https://github.com/samtools/samtools

REPOSITORY

https://github.com/samtools/samtools

SOURCE LINK

Command

tabix

```
tabix -p vcf {output.vcf}
```

6 Count reads with REF vs. ALT allele in a sample for each SNP.

Software

gatk NAME

Command

ASEReadCounter

```
gatk ASEReadCounter \
    -R {input.ref} \
    -I {input.bam} \
    -V {input.vcf} \
    -O {output} \
    --min-depth-of-non-filtered-base 10 \
    --min-mapping-quality 250 \
    --min-base-quality 15
```

7 Compare similarity of genotypes and allele counts from RNA-Seq to identify mixups.

qc_rna_to_geno_similarity.py

Produces a matrix of RNA-Seq samples x VCF individuals giving similarity across the test SNPs. Similarity is mean(1 - abs(RNA_frac - geno_frac)), where RNA_frac is fraction of reads with ALT allele for each SNP, and geno_frac is fraction of DNA strands with ALT allele (i.e. 0, 0.5, or 1) for each SNP. Also produces a summary table with top similarity per RNA-Seq sample to quickly check for mismatches.

8 For samples without a genotype match, check for matches in all rats.

qc_rna_to_geno_all_rats.py

After running the previous step, some mismatched RNA samples might still not have a genotype match. This rule will compare their test SNPs to those of all 6000+ rats we have genotypes for to see if any match. It's good to also include at least one RNA sample ID that did match as a positive control (as long as it's included in the all-rat VCF).

9 Examine the outputs to identify samples that need to be relabeled (e.g. if two labels get swapped) or removed.

- To relabel a sample, edit the ID in the 2nd column of fastq_map.txt for all of its FASTQ files so that its BAM file gets labeled correctly. You'll then need to regenerate the BAM file since it will now use the correct VCF individual as input to STAR.
- To remove a sample, remove its ID from rat_ids.txt and delete its BAM and any other generated files.

If a match is found with a rat outside the current dataset in the previous step, you'll need to add the new matching genotypes to the VCF file. However, if the matching genotypes differ greatly from the current dataset, e.g. they include a very different set of SNPs, it may be better to just remove the sample.

Quantify gene expression

10 Generate the index for RSEM.

Software

RSEM NAME

Command

rsem-prepare-reference

```
rsem-prepare-reference \
    {input.fasta} \
    ref/rsem_reference/rsem_reference \
    --gtf {input.gtf} \
    --num-threads {resources.cpus}
```

11 Quantify expression from a BAM file.

Software

RSEM NAME

Command

rsem-calculate-expression

```
rsem-calculate-expression \
    {params.paired_end_flag} \
    --num-threads {resources.cpus} \
    --quiet \
    --estimate-rspd \
    --no-bam-output \
    --alignments {input.bam} \
    {params.ref_prefix} \
    {params.out_prefix}
```

Software

NAME

gzip

Command

gzip

```
gzip {params.out_prefix}.genes.results
```

12 Combine all isoforms of a gene into a single transcript.

collapse_annotation.py

Originally used in the **GTEx pipeline**.

Command

new command name

```
python3 src/collapse_annotation.py \
    ref/Rattus_norvegicus.Rnor_6.0.99.gtf \
    ref/Rattus_norvegicus.Rnor_6.0.99.genes.gtf
```

13 Combine RSEM output from all samples into log-count and TPM expression tables.

assemble_expression.py

Also computes inverse-quantile normalized values to use for eQTL mapping. The filtered version is to avoid a tensorQTL error on phenotypes with 1 nonzero value.

Command

new command name

```
python3 src/assemble_expression.py {params.rsem_dir} {input.anno}
{params.prefix}
```

Software

samtools

NAME

https://github.com/samtools/samtools

REPOSITORY

https://github.com/samtools/samtools

SOURCE LINK

Command

bgzip

```
bgzip {params.prefix}.log2.bed
bgzip {params.prefix}.tpm.bed
bgzip {params.prefix}.iqn.bed
bgzip {params.prefix}.iqn.filtered.bed
```

Command

tabix

```
tabix {params.prefix}.log2.bed.gz
tabix {params.prefix}.tpm.bed.gz
tabix {params.prefix}.iqn.bed.gz
tabix {params.prefix}.iqn.filtered.bed.gz
```

Map eQTLs

14 Get SNPs with sufficient variation in a given set of samples and convert VCF to plink.

Software

plink

NAME

https://www.cog-genomics.org/plink/2.0/

SOURCE LINK

Command

plink2 make bed

```
plink2 --make-bed \
    --vcf {input.vcf} \
    --keep {input.samples} \
    --maf 0.01 \
    --mac 2 \
    --max-alleles 2 \
    --out {params.prefix}
```

15 Prune genotypes to compute covariate PCs.

--indep-pairwise parameters are based on GTEx methods.

Software

plink

NAME

https://www.cog-genomics.org/plink/2.0/

SOURCE LINK

Command

plink2 prune

```
plink2 \
    --bfile {params.prefix} \
    --geno 0.05 \
    --maf 0.05 \
    --indep-pairwise 200 100 0.1 \
    --out {params.pruned_prefix}
```

Command

plink2 subset to pruned

```
plink2 \
    --bfile {params.prefix} \
    --extract {params.pruned_prefix}.prune.in \
    --export vcf bgz id-paste=iid \
    --out {params.pruned_prefix}
```

16 Compute genotype (n=5) and expression (n=20) PCs and combine into covariates table.

covariates.R

Command

new command name

```
Rscript src/covariates.R {input.vcf} {input.bed} {params.n_geno_pcs}
{params.n_expr_pcs} {output}
```

17 Map cis-eQTLs, determining significance using permutations.

Outputs the top association per gene. This script calls tensorQTL and uses the random_tiebreak parameter, which is important for outbred populations in which there are often multiple top eSNPs in 100% LD. Without the random tiebreak, the first or last tied top SNP is returned, resulting in positional bias.

Software

tensorQTL NAME

run_tensorqtl.py

Command

new command name

```
python3 src/run_tensorqtl.py \
    {params.geno_prefix} \
    {input.bed} \
    {output} \
    --covariates {input.covar} \
    --mode cis
```

18 Use stepwise regression to identify multiple conditionally independent cis-eQTLs per gene.

Software

tensorQTL NAME

Command

new command name

```
python3 src/run_tensorqtl.py \
    {params.geno_prefix} \
    {input.bed} \
    {output} \
    --covariates {input.covar} \
    --cis_output {input.cis} \
    --mode cis_independent
```

19 Get summary statistics for all tested cis-window SNPs per gene.

Software

tensorQTL NAME

Command

tensorqtl cis_nominal

```
python3 -m tensorqtl \
    {params.geno_prefix} \
    {input.bed} \
    {wildcards.tissue} \
    --covariates {input.covar} \
    --output_dir {params.outdir} \
    --mode cis_nominal
```

20 Map trans-eQTLs.

Software

tensorQTL NAME

Command

tensorqtl trans

```
python3 -m tensorqtl \
    {params.geno_prefix} \
    {input.bed} \
    {wildcards.tissue} \
    --covariates {input.covar} \
    --output_dir {params.outdir} \
    --output_text \
    --batch_size 10000 \
    --mode trans
```

21 Extract all significant cis SNP-gene pairs.

tensorqtl_all_signif.py

Command

new command name

```
python3 src/tensorqtl_all_signif.py {input.perm} {params.nom_prefix}
{output}
```

22 Extract p-values for all tested cis-window SNPs per gene.

tensorqtl_all_cis_pvals.py

Command

new command name

```
python3 src/tensorqtl_all_cis_pvals.py {params.nom_dir} {output}
```

23 Get effect size (allelic fold change) for top association per gene and all significant cis-eQTLs.

Software

aFC NAME

https://github.com/secastel/aFC REPOSITORY

prepare_qtl_for_afc.py

Command

new command name

```
python3 tools/aFC/aFC.py \
    --vcf {input.vcf} \
    --pheno {input.bed} \
    --qtl <(python3 src/prepare_qtl_for_afc.py {input.qtl}
{input.qtl_indep}) \
    --cov {input.covar} \
    --log_xform 1 \
    --output {output}
```