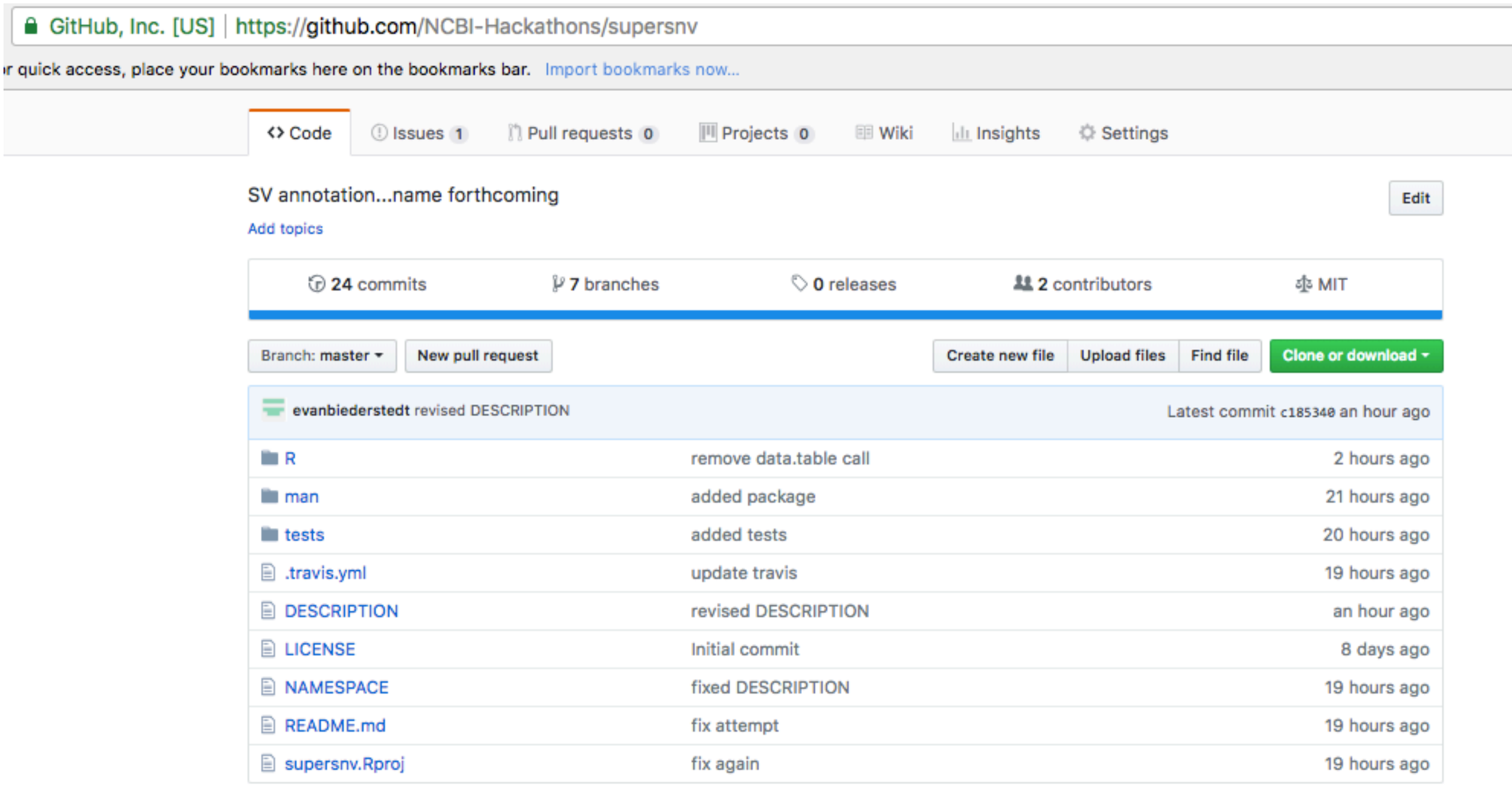
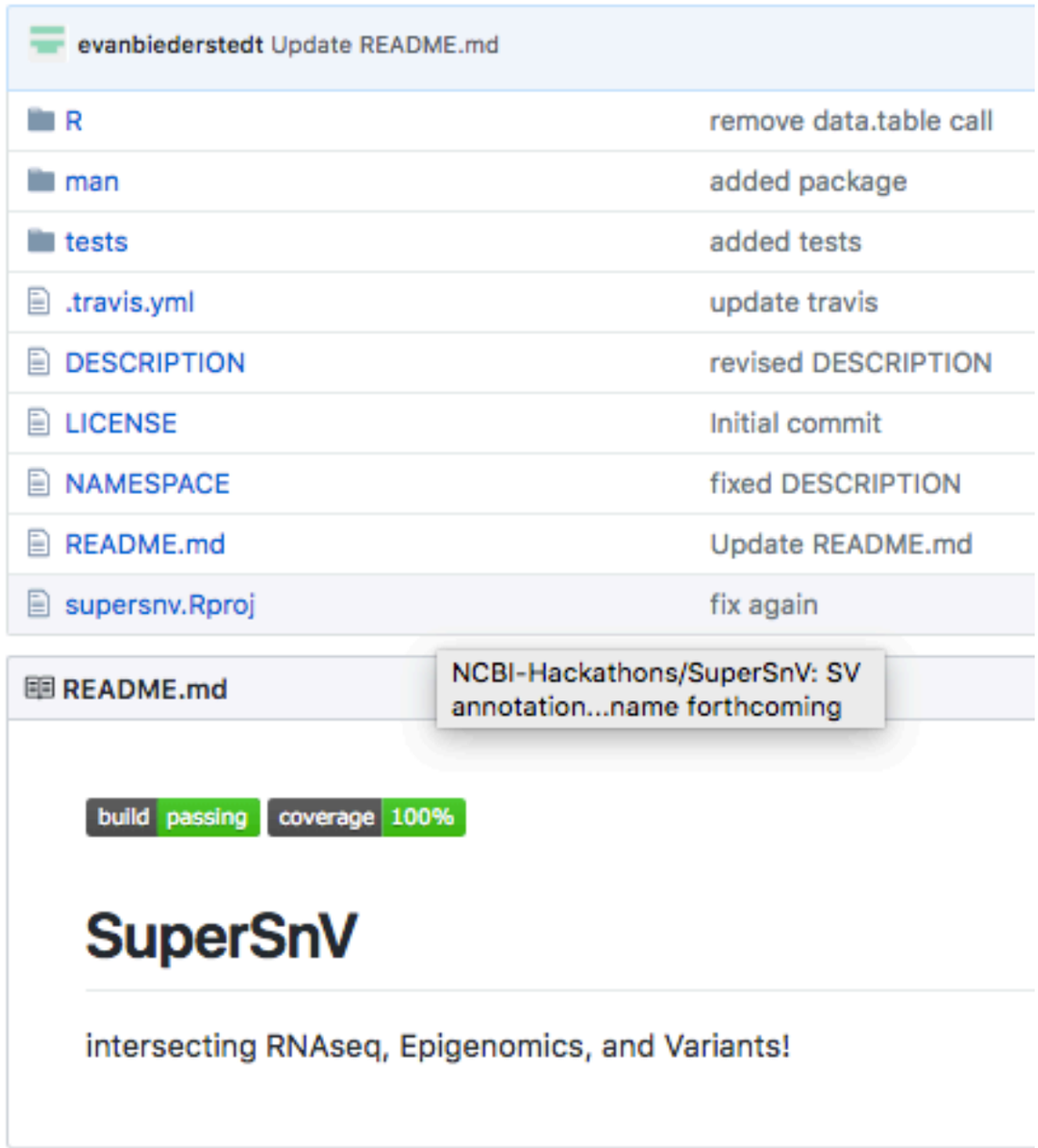
Day 2

SV annotation!

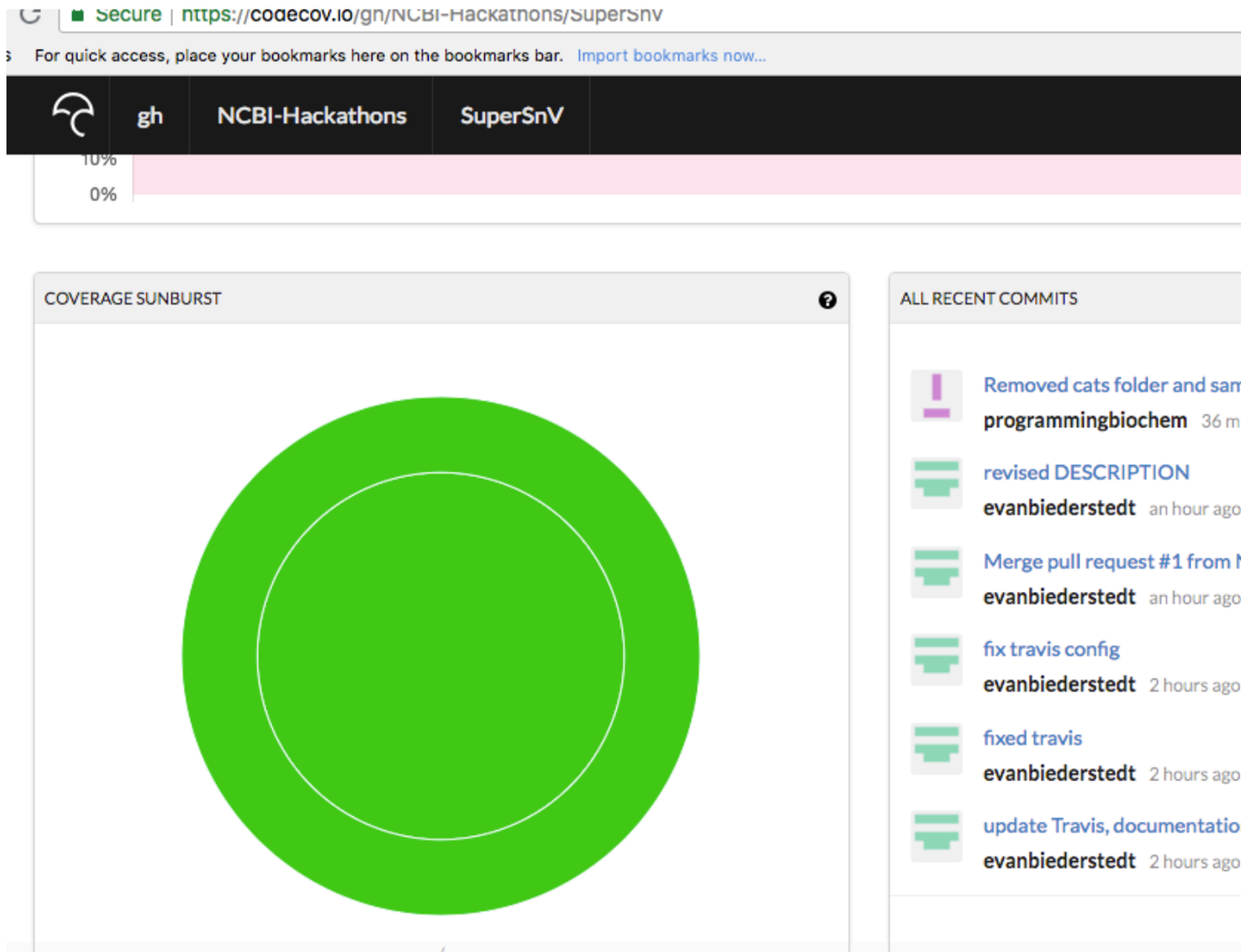
(We need a better name…)



Milestones!



- **working R package!**
- **unit testing!**
- **Travis-CI tests**



Goal: Build R package to annotate SV BEDPEs and VCFs

Inputs:
SV VCF/ BEDPE File
Window Size (length)
Reference genome (hg19/ hg38)
Annotations:
Gencode, Ensembl
Input data: GTEx RNAseq data

→

R package: using GRanges, data.table, annotatedb

Outputs:
Annotated VCF/Bed PE file with affected Gene Expression data

Challenges: translocations? Standards? Gene fusions? Features?

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

SuperSnV notes

Create an R package containing helper functions to annotate variants from a VCF/BEDPE-format file. The annotation could be done based on the hg19/hg38 human reference genome build, and the annotation release from either GENCODE (https://www.gencodegenes.org/releases/19.html) or ENSEMBL (https://useast.ensembl.org/info/data/ftp/index.html) consortium.

Potential input parameters for the main function():

- File = VCF/BEDPE format file [must specify or exit with error]
- Reference genome build: hg19/hg38 [default?]
- Annotation: GENCODE/ENSEMBL [default?]
- RNASeq : GTeX/ user's RNASeq [optional, default?]
- Window size= in base pairs (bp) for SV. [default?]
- Backwards/ left flanking region = in base pairs [default?]
- Forwards/right flanking region = in base pairs [default?]
- Output = path to output file (should this also be in bedpe/vcf format?)

Main function()

If VCF, there are no SVs but only SNPs

- Read in VCF as Granges
- Make sure works for hg19/hg38
- findOverlaps() annotate SNPs with gene names
- add column for gene expression based on the annotated gene names
- make sure rows double???

If BEDPE, use data.table

- make sure file format is valid
- classify if structural variants are deletion/duplication/inversion/translocation?
- if chr1==chr2
    - translate to Granges
    - annotated columns for gene names

  elseif chr1 != chr2

  this is a translocation, and then determine where the overlap happens,

  annotate with overlapping genes

Potential helper functions

For annotation files

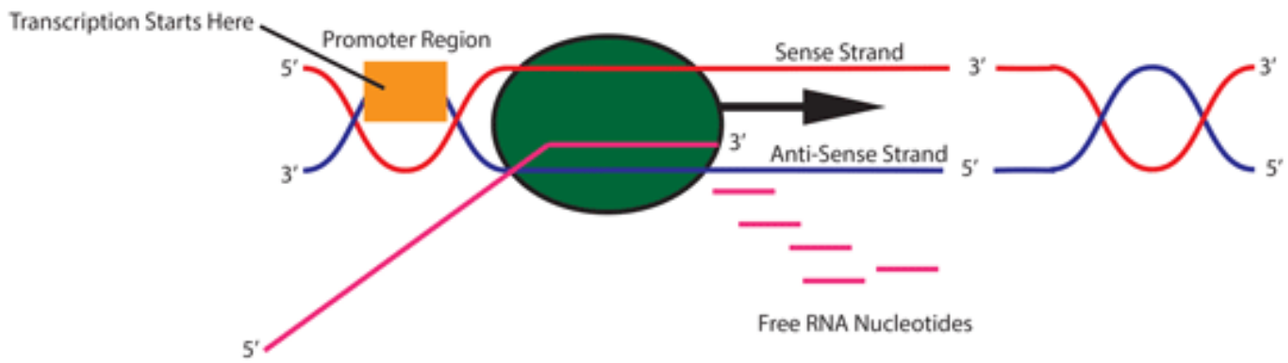
- GTFValidator
- Check which annotation to use (GENCODE/ENSEMBL)?
- Check genome build (R tool should support hg19 and hg38 reference genome build)

```
source("https://bioconductor.org/biocLite.R")
biocLite("GenomicInteractions")
biocLite("GenomicRanges")
library('GenomicInteractions')
library("GenomicRanges")
library("rtracklayer")


validate_bedpe_file <- function(file) {
    input <- as.data.frame(import(file,format='bedpe'))
    #----check-length----#
    input_len <- length(input)
    if (input_len < 10) {
        print ("length of the input BEDPE has less than 10 columns!")}
        result <- "True"
    #----check-typeof for the first 10 rows ----#
for(i in 1:10) {
    if (typeof(t(input)[1, i])!='character') {
        result <- "False"
    }
    if (as.integer(t(input)[2, i])) {
        result <- "False"
    }
}

```

Annotating the SNVs and SVs by the affected Transcriptional Promoter Regions

Transcriptional Promoter Regions From FANTOM5

Experimentally Validated: 273790 promoters for 32342 genes



Gene Annotations From GENCODE

Preprocessing & Matching

237,989 Promoter Regions for 19,964 Genes

```python
file_in = open('RAW_hg19.cage_peak_ann.txt', 'r')
file_out = open('Hackathon_Peng_Promoter.txt', 'w')

total_count = 0
gene_count = 0
gene_set = set()

for eachline in file_in:
    if eachline.startswith('chr'):
        component = eachline.strip().split('\t')
        annotation = component[0]
        description = component[1]
        uniprot = component[6]

        description_info = description.split('@')
        if description_info[0] != 'p':
            gene = description_info[1]
            annotation_split_1 = annotation.split(':')
            chrom = annotation_split_1[0]
            annotation_split_2 = annotation_split_1[1].split('..')
            start = annotation_split_2[0]
            annotation_split_3 = annotation_split_2[1].split(',')
            end = annotation_split_3[0]
            strand = annotation_split_3[1]

            if not gene.startswith('ENST'):
                total_count += 1
                gene_set.add(gene)
                file_out.write(chrom + '\t' + start + '\t' + end + '

gene_count = len(gene_set)
print 'TOTAL COUNT:\t' + str(total_count)
print 'GENE COUNT:\t' + str(gene_count)
```

```python
file_in_gene = open('Hackathon_Peng_Genename.txt', 'r')
file_in_promoter = open('Hackathon_Peng_Promoter.txt', 'r')

file_out = open('Hackathon_Peng_Promoter_matched.txt', 'w')

gene_set = set()
for eachline in file_in_gene:
    gene_set.add(eachline.strip())

total_count = 0
gene_match_set = set()
promoter_head = file_in_promoter.readline()
file_out.write(promoter_head)
for eachline in file_in_promoter:
    component = eachline.strip().split('\t
    gene = component[4]
    if gene in gene_set:
        file_out.write(eachline)
        total_count += 1
        gene_match_set.add(gene)

gene_match_count = len(gene_match_set)
print 'TOTAL COUNT:\t' + str(total_count)
print 'GENE COUNT:\t' + str(gene_match_cou
```

chr10	100008587	100008589	+	CU680531
chr10	100015362	100015397	-	LOXL4
chr10	100017518	100017519	-	LOXL4
chr10	100027943	100027958	-	LOXL4
chr10	100028159	100028160	-	LOXL4
chr10	100151376	100151377	-	PYROXD2
chr10	100171201	100171209	-	PYROXD2
chr10	100174900	100174956	-	PYROXD2
chr10	100174957	100174982	-	PYROXD2
chr10	100179836	100179849	-	HPS1
chr10	100179866	100179875	-	HPS1
chr10	100185639	100185659	-	HPS1
chr10	100191028	100191047	-	HPS1
chr10	100202961	100202987	-	HPS1