HYENA User Manual

Version 0.5.4

1 Introduction

HYENA (Hijacking of Enhancer Activity) is a computational tool to detect genes activated by nearby somatic structural variations (SVs), a phenomenon known as enhancer hijacking. It takes somatic SVs and gene expression data as inputs and outputs enhancer hijacking candidate genes. This vignette describes the components of HYENA with an example pipeline that uses Pan-Cancer Analysis of Whole Genomes (PCAWG) thyroid cancer (THCA) dataset.

2 Prerequisites and input files

2.1 Required packages

R 4.0.3 and the following list of R packages are required to run HYENA properly:

- optparse 1.6.6 or above
- data.table 1.14.2 or above
- stringr 1.4.0 or above
- stats 4.0.3 or above
- MatrixGenerics 1.2.0 or above
- ggplot2 3.3.5 or above
- RMThreshold 1.1 or above
- tidyr 1.1.2 or above
- GenomicRanges 1.42.0 or above
- IRanges 2.24.1 or above
- dplyr 1.0.7 or above
- S4Vectors 0.28.1 or above
- lattice 0.22.44 or above
- preprocessCore 1.52.1 or above
- EDASeq 2.24.0 or above

All R packages can be installed within a few minutes. For package installation issues, please contact your system admin.

2.2 Testing data set and an example pipeline

The user can run the following code to execute the entire pipeline on the THCA example:

```
cd example
sh ./example.sh
```

This will run HYENA using 0 to 5 principal components in the normal-score regression model while calculating empirical p-values for each run by permuting gene expression data 5 times. The final results for each run will be under "./example/results/" within their respective run folders (e.g. "run_PC0", "run_PC1", etc.). The user should be able to generate "DAT.annot_sv_pur_cn_age_sex_PC0_posEstimate_adj_sig.txt" and "DAT.annot_sv_pur_ cn_age_sex_PC0_posEstimate_adj_sig_rmeqtl.txt" files. The number of significant genes and p-values may be slightly different from the files we provide. The example pipeline can be completed in a few minutes on a normal desktop.

If the user is unable to run HYENA with all the packages properly installed, or HYENA runs smoothly but the results are dramatically different from the files we provide, it is likely due to the specific version of the R packages the user is loading.

2.3 Overview of the pipeline

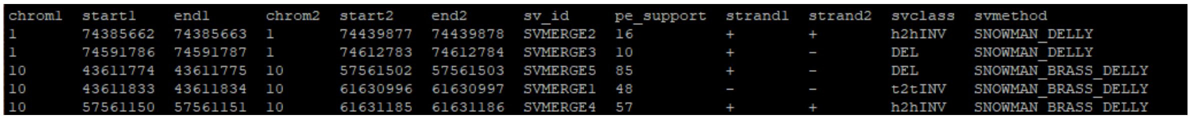
HYENA is a multi-step pipeline. STEP 1 involves creating transcription start site (TSS) windows for each gene where SV breakpoints will be mapped based on a reference genome. STEP 2 is where the breakpoint locations from each SV are mapped to these TSS windows. STEP 3 prepares expression data by normalizing them across samples, adding a small Gaussian noise, calculating normal scores, and performing principal component analysis. STEP 4 performs normal-score regression on observed regression and identifies candidate genes. STEP 5 calculates empirical p-values by permutation. Finally, STEP 6 looks for associations between known eQTLs of the candidate genes and their SV genotypes to determine whether the observed expression changes can be explained by eQTLs. **Users can choose to end the analysis afterthe normal-score regression step (STEP 4), but we highly recommend running STEP 5 for more reliable results.**

2.4 Input file format

Note that empty lines are not allowed in any input files.

There are three input files that are essential to the HYENA pipeline: SV calls, gene expression data, and sample/aliquot IDs. Individual bedpe files for each sample, a file containing expression data from all samples in a single matrix, and a file with sample IDs are the bare minimum input files. **For best performance, the users should include gene level copy number, sample purity, and clinical data with patient age and sex information to be modeled as well.**

The SV data should be provided as individual bedpe files and named as “sampleid.bedpe”. **For samples with no SVs, do not generate empty bedpe files.** The contents of each file should follow the standard bedpe format as shown in Figure 1 where the first three columns are the genomic location of the first breakpoint of an SV and the following three columns are the genomic location of the second breakpoint for the same SV. The smaller coordinate should be “start1” and the larger one should be “start2”. “strand1” and “strand2” indicate breakpoint orientation. For example, a deletion should have “+” and “-”, whereas a tandem duplication should be "-" and "+". **Note that the strand annotation may be different for different SV callers and it’s important to annotation strands according to PCAWG definition (del: +-, etc.).**The column order should be exactly in the indicated order. All other columns are not necessary, but the columns should be present and filled with any string.

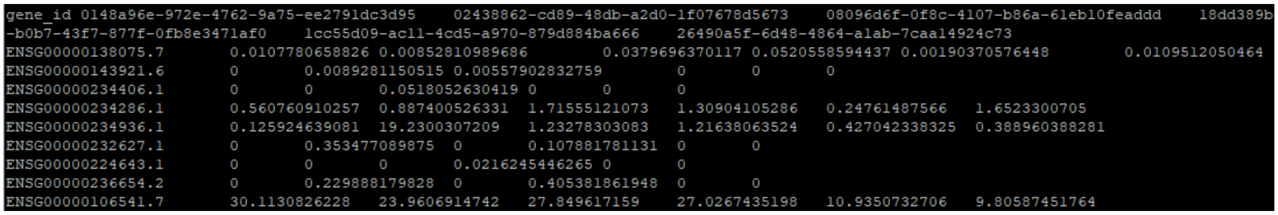


```
chrom1  start1    end1      chrom2  start2    end2      sv_id     pe_support  strand1  strand2  svclass  svmethod
1       74385662  74385663  1       74439877  74439878  SVMERGE2  16          +        +        h2hINV   SNOWMAN_DELLY
1       74591786  74591787  1       74612783  74612784  SVMERGE3  10          +        -        DEL      SNOWMAN_DELLY
10      43611774  43611775  10      57561502  57561503  SVMERGE5  85          +        -        DEL      SNOWMAN_BRASS_DELLY
10      43611833  43611834  10      61630996  61630997  SVMERGE1  48          -        -        t2tINV   SNOWMAN_BRASS_DELLY
10      57561150  57561151  10      61631185  61631186  SVMERGE4  57          +        +        h2hINV   SNOWMAN_BRASS_DELLY
```

Figure 1: SV data format (bedpe)

Note that the quality of input SVs is very important. We recommend using multiple SV callers. In our experience, many SV callers produce caller-specific artifacts. The artifacts are often small deletions, small inversions or translocations. They typically have few read support and/or overlap with repetitive sequences. We recommend using approach described in our paper (PMID: 32813322) to identify SV artifacts.

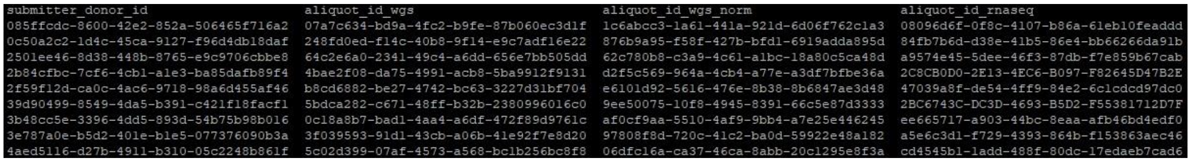
The gene expression matrix should be provided in the format depicted in Figure 2 with gene ID as row names and sample ID as column names. HYENA takes any forms of gene expression quantification as input, such as FPKM, TPM, or other normalized values. The example given in Figure 2 is FPKMs from the PCAWG study. The expression matrix can have 0s to indicate no expression for a particular gene. However, **“NA”s or empty cells are not allowed**. Gene IDs should be the same across all input and reference files and the sample IDs should match aliquot_id_rnaseq in Sample_ID file.



```
gene_id 0148a96e-972e-4762-9a75-ee2791dc3d95    02438862-cd89-48db-a2d0-1f07678d5673    08096d6f-0f8c-4107-b86a-61eb10feaddd    18dd389b
-b0b7-43f7-877f-0fb8e3471af0    1cc55d09-ac11-4cd5-a970-879d884ba666    26490a5f-6d48-4864-a1ab-7caa14924c73
ENSG00000138075.7       0.0107780658826 0.00852810989686        0.0379696370117 0.0520558594437 0.00190370576448        0.0109512050464
ENSG00000143921.6       0       0.0089281150515 0.00557902832759        0       0       0
ENSG00000234406.1       0       0       0.0518052630419 0       0       0
ENSG00000234286.1       0.560760910257  0.887400526331  1.71555121073   1.30904105286   0.24761487566   1.6523300705
ENSG00000234936.1       0.125924639081  19.2300307209   1.23278303083   1.21638063524   0.427042338325  0.388960388281
ENSG00000232627.1       0       0.353477089875  0       0.107881781131  0       0
ENSG00000224643.1       0       0       0       0.0216245446265 0       0
ENSG00000236654.2       0       0.229888179828  0       0.405381861948  0       0
ENSG00000106541.7       30.1130826228   23.9606914742   27.849617159    27.0267435198   10.9350732706   9.80587451764
```

Figure 2: Gene expression matrix format

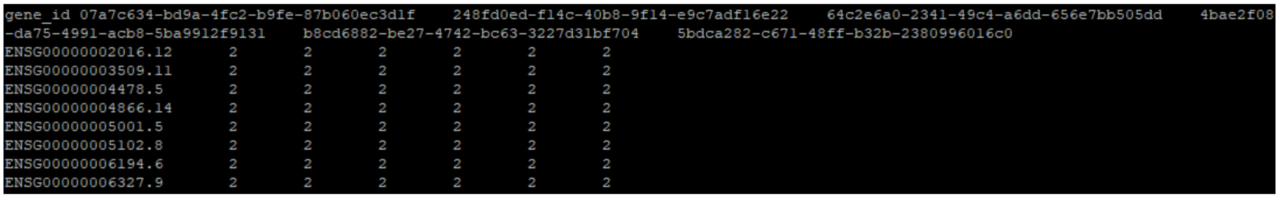
Sample-ID file serves two purposes for HYENA: (1) It provides the list of samples that the user would like to analyze. **If the user would like to run only a subset of the samples, they can do so just by limiting their list of sample IDs in the "sample_ids.txt" file without subsetting other input files.** (2) It facilitates mapping of different sample aliquot IDs from a single donor/patient. In studies where the donor and the samples collected are identified with the same unique ID, matching the corresponding RNA and DNA samples may not be an issue. However, in most comprehensive multi-omic studies such as PCAWG, International Cancer Genome Consortium (ICGC), and The Cancer Genome Atlas (TCGA), RNA and DNA samples have different aliquot numbers are different from donor IDs. Therefore, the "sample_ids.txt" file is formatted in four columns as depicted in Figure 3: "submitter_donor_id", "aliquot_id_wgs", "aliquot_id_wgs_norm" (ID for the matched normal sample, this column is required for eQTL analysis) and "aliquot_id_rnaseq". This allows HYENA to identify the correct whole genome sequencing data (used for SV calls, copy number calls etc.) and RNA sequencing data. **Samples listed in the sample-ID file must be in the gene expression matrix.** If all IDs are the same, the user can simply populate all columns with the same values. **Note that some special characters in sample IDs may cause the algorithm to crash.**



```
submitter_donor_id                    aliquot_id_wgs                        aliquot_id_wgs_norm                   aliquot_id_rnaseq
085ffcdc-8600-42e2-852a-506465f716a2  07a7c634-bd9a-4fc2-b9fe-87b060ec3d1f  1c6abcc3-1a61-441a-921d-6d06f762c1a3  08096d6f-0f8c-4107-b86a-61eb10feaddd
0c50a2c2-1d4c-45ca-9127-f96d4db18daf  248fd0ed-f14c-40b8-9f14-e9c7adf16e22  876b9a95-f58f-427b-bfd1-6919adda895d  84fb7b6d-d38e-41b5-86e4-bb66266da91b
2501ee46-8d38-448b-8765-e9c9706cbbe8  64c2e6a0-2341-49c4-a6dd-656e7bb505dd  62c780b8-c3a9-4c61-a1bc-18a80c5ca48d  a9574e45-5dee-46f3-87db-f7e859b67cab
2b84cfbc-7cf6-4cb1-a1e3-ba85dafb89f4  4bae2f08-da75-4991-acb8-5ba9912f9131  d2f5c569-964a-4cb4-a77e-a3df7bfbe36a  2C8CB0D0-2E13-4EC6-B097-F82645D47B2E
2f59f12d-ca0c-4ac6-9718-98a6d455af46  b8cd6882-be27-4742-bc63-3227d31bf704  e6101d92-5616-476e-8b38-8b6847ae3d48  47039a8f-de54-4ff9-84e2-6c1cdcd97dc0
39d90499-8549-4da5-b391-c421f18facf1  5bdca282-c671-48ff-b32b-2380996016c0  9ee50075-10f8-4945-8391-66c5e87d3333  2BC6743C-DC3D-4693-B5D2-F55381712D7F
3b48cc5e-3396-4dd5-893d-54b75b98b016  0c18a8b7-bad1-4aa4-a6df-472f89d9761c  af0cf9aa-5510-4af9-9bb4-a7e25e446245  ee665717-a903-44bc-8eaa-afb46bd4edf0
3e787a0e-b5d2-401e-b1e5-077376090b3a  3f039593-91d1-43cb-a06b-41e92f7e8d20  97808f8d-720c-41c2-ba0d-59922e48a182  a5e6c3d1-f729-4393-864b-f153863aec46
4aed5116-d27b-4911-b310-05c2248b861f  5c02d399-07af-4573-a568-bc1b256bc8f8  06dfc16a-ca37-46ca-8abb-20c1295e8f3a  cd4545b1-1add-488f-80dc-17edaeb7cad6
```

Figure 3: List of sample IDs

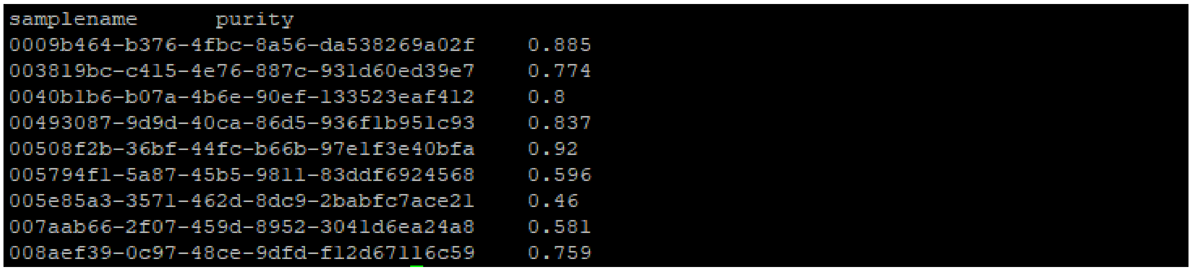
HYENA also incorporates copy number data. The users can feed integer copy number calls for each gene in order to mitigate the effect of differential gene expression due to copy number differences between samples. The copy number input file has similar format as expression data as shown in Figure 4. The sample IDs should be aliquo_id_wgs from Sample-ID file. NAs are allowed in copy number data.



```
gene_id 07a7c634-bd9a-4fc2-b9fe-87b060ec3d1f    248fd0ed-f14c-40b8-9f14-e9c7adf16e22    64c2e6a0-2341-49c4-a6dd-656e7bb505dd    4bae2f08
-da75-4991-acb8-5ba9912f9131    b8cd6882-be27-4742-bc63-3227d31bf704    5bdca282-c671-48ff-b32b-2380996016c0
ENSG00000002016.12      2       2       2       2       2       2
ENSG00000003509.11      2       2       2       2       2       2
ENSG00000004478.5       2       2       2       2       2       2
ENSG00000004866.14      2       2       2       2       2       2
ENSG00000005001.5       2       2       2       2       2       2
ENSG00000005102.8       2       2       2       2       2       2
ENSG00000006194.6       2       2       2       2       2       2
ENSG00000006327.9       2       2       2       2       2       2
```

Figure 4: Absolute copy number calls for each gene in WGS samples

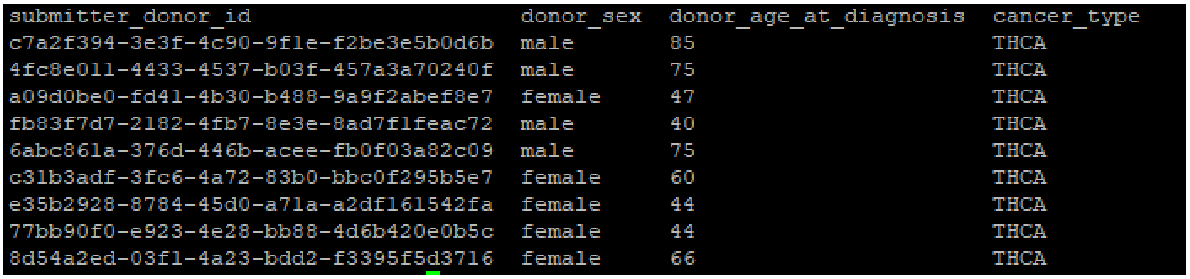
In studies where sample purity is available, users can choose to correct gene expression for sample purity by providing a file with a "samplename" (WGS) column and a "purity" column (Figure 5). Purity should be values between 0 and 1.



```
samplename      purity
0009b464-b376-4fbc-8a56-da538269a02f    0.885
003819bc-c415-4e76-887c-931d60ed39e7    0.774
0040b1b6-b07a-4b6e-90ef-133523eaf412    0.8
00493087-9d9d-40ca-86d5-936f1b951c93    0.837
00508f2b-36bf-44fc-b66b-97e1f3e40bfa    0.92
005794f1-5a87-45b5-9811-83ddf6924568    0.596
005e85a3-3571-462d-8dc9-2babfc7ace21    0.46
007aab66-2f07-459d-8952-3041d6ea24a8    0.581
008aef39-0c97-48ce-9dfd-f12d67116c59    0.759
```

Figure 5: Sample purity information to be used in the linear model.

Finally, users can choose to incorporate clinical data. For now, HYENA only supports donor sex, age, and cancer type. If users would like to correct for these confounding factors in the regression model, they can provide a clinical data file with four columns: “submitter_donor_id”, “donor_sex”, “donor_age_at_diagnosis”, and “cancer_type” (Figure 6). The order of these columns does not matter.



```
submitter_donor_id                    donor_sex  donor_age_at_diagnosis  cancer_type
c7a2f394-3e3f-4c90-9f1e-f2be3e5b0d6b  male       85                      THCA
4fc8e011-4433-4537-b03f-457a3a70240f  male       75                      THCA
a09d0be0-fd41-4b30-b488-9a9f2abef8e7  female     47                      THCA
fb83f7d7-2182-4fb7-8e3e-8ad7f1feac72  male       40                      THCA
6abc861a-376d-446b-acee-fb0f03a82c09  male       75                      THCA
c31b3adf-3fc6-4a72-83b0-bbc0f295b5e7  female     60                      THCA
e35b2928-8784-45d0-a71a-a2df161542fa  female     44                      THCA
77bb90f0-e923-4e28-bb88-4d6b420e0b5c  female     44                      THCA
8d54a2ed-03f1-4a23-bdd2-f3395f5d3716  female     66                      THCA
```

Figure 6: Format of the clinical meta data for each sample

3 Workflow

3.1 Step 1: Calculating TSS windows

A list of TSS-flanking windows needs to be generated using “calc_wind.R”. This script will start with a desired reference genome file and create a file with four new columns that define the TSS window for each gene: (1) “tss”, transcription start site; (2) “tts”, transcription termination site; (3) “tss_left”, the left boundary of the TSS window; (4) “tss_right”, right boundary of the TSS window.

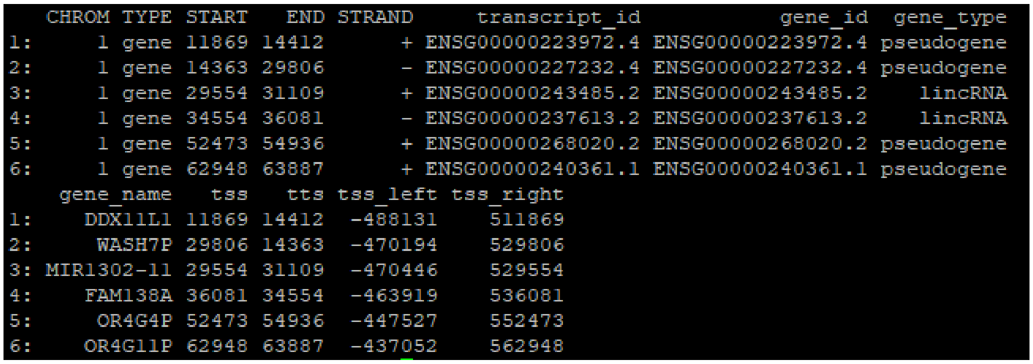
Arg	Description
-r, --ref	Gene annotation
-u, --up	Number of basepairs UPSTREAM of transcription start site (default: 500000)
-d, --down	Number of basepairs DOWNSTREAM of transcription start site (default: 500000)
-w, --write	Output file (default: "./tss_windows.txt")
-v, --verbose	Verbose mode for troubleshooting (default: FALSE)

Table 1: Description of arguments for calc_wind.R

Table 1 shows the arguments available for “calc_wind.R”. In the THCA example, this can be done with the following command:

```
Rscript ../R/calc_wind.R \
-r ../ref/toy_gene_annot_hg19.txt \
-w ../ref/toy_gene_annot_hg19_tsswindow.txt
```

The example pipeline uses "toy_gene_annot_hg19.txt" for demonstration purposes. Users should not use the toy annotation file other than the test run, but instead use "gene_annot_hg19.txt" and "gene_annot_hg38.txt" provided for actual analysis. Running the above code will generate a file



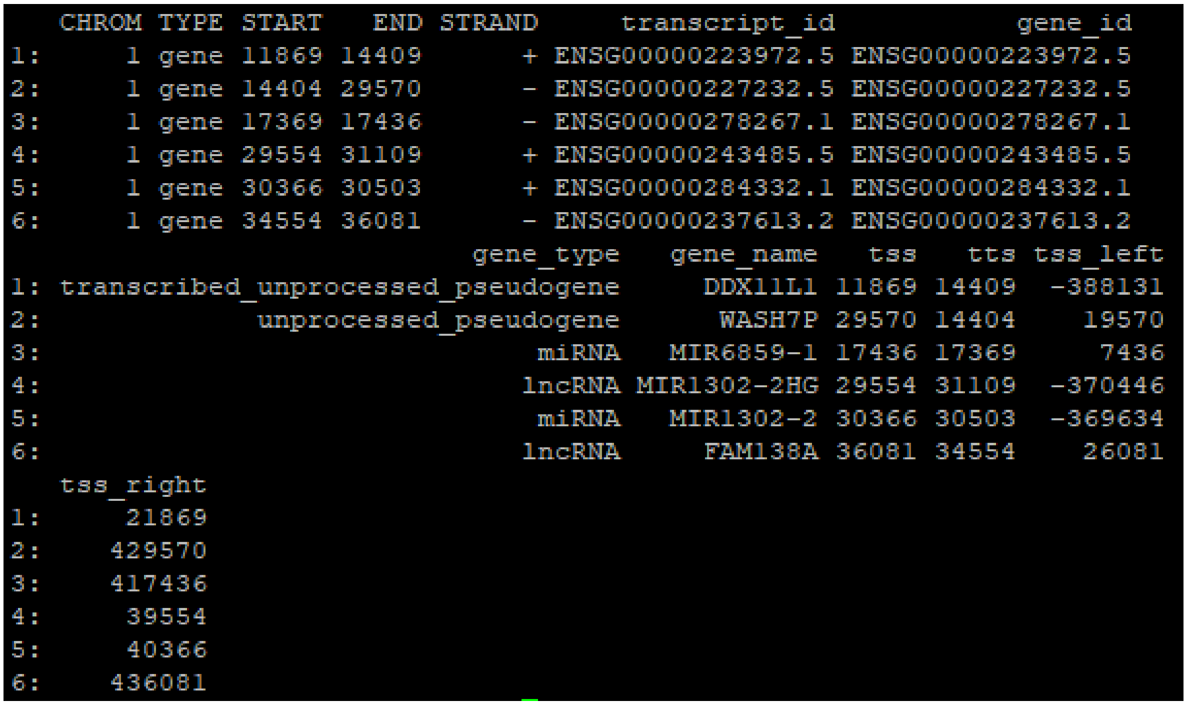
```
   CHROM TYPE START   END STRAND     transcript_id           gene_id  gene_type
1:     1 gene 11869 14412      + ENSG00000223972.4 ENSG00000223972.4 pseudogene
2:     1 gene 14363 29806      - ENSG00000227232.4 ENSG00000227232.4 pseudogene
3:     1 gene 29554 31109      + ENSG00000243485.2 ENSG00000243485.2    lincRNA
4:     1 gene 34554 36081      - ENSG00000237613.2 ENSG00000237613.2    lincRNA
5:     1 gene 52473 54936      + ENSG00000268020.2 ENSG00000268020.2 pseudogene
6:     1 gene 62948 63887      + ENSG00000240361.1 ENSG00000240361.1 pseudogene
    gene_name   tss   tts tss_left tss_right
1:    DDX11L1 11869 14412  -488131    511869
2:     WASH7P 29806 14363  -470194    529806
3: MIR1302-11 29554 31109  -470446    529554
4:    FAM138A 36081 34554  -463919    536081
5:     OR4G4P 52473 54936  -447527    552473
6:    OR4G11P 62948 63887  -437052    562948
```

named “toy_gene_annot_hg19_tsswindow.txt” (Figure 7). Notice that for genes that are close to the beginning of the chromosome, left boundary can be a negative basepair location. Similarly, for genes that are close tothe end of the chromosome, the right boundary may extend beyond the length of the chromosome. This will not cause any issues moving forward.

Figure 7: TSS windows spanning 500Kb on either side of the TSS for each gene in "hg19"

The windows with respect to TSS sites can be customized. For example, if we wanted to use the hg38 reference genome and consider 400Kb upstream and 10kb downstream of TSS, we would run the command below and create “gene_annot_hg38_tsswindow.txt” (Figure 8).

```
Rscript ../R/calc_wind.R \
-r ../ref/gene_annot_hg38.txt \
-w ../ref/gene_annot_hg38_tsswindow.txt \
-u 400000 -d 10000
```



```
   CHROM TYPE START   END STRAND      transcript_id           gene_id
1:     1 gene 11869 14409      + ENSG00000223972.5 ENSG00000223972.5
2:     1 gene 14404 29570      - ENSG00000227232.5 ENSG00000227232.5
3:     1 gene 17369 17436      - ENSG00000278267.1 ENSG00000278267.1
4:     1 gene 29554 31109      + ENSG00000243485.5 ENSG00000243485.5
5:     1 gene 30366 30503      + ENSG00000284332.1 ENSG00000284332.1
6:     1 gene 34554 36081      - ENSG00000237613.2 ENSG00000237613.2
                          gene_type   gene_name   tss   tts tss_left
1: transcribed_unprocessed_pseudogene     DDX11L1 11869 14409  -388131
2:             unprocessed_pseudogene      WASH7P 29570 14404    19570
3:                              miRNA   MIR6859-1 17436 17369     7436
4:                             lncRNA MIR1302-2HG 29554 31109  -370446
5:                              miRNA   MIR1302-2 30366 30503  -369634
6:                             lncRNA     FAM138A 36081 34554    26081
   tss_right
1:     21869
2:    429570
3:    417436
4:     39554
5:     40366
6:    436081
```

Figure 8: Customized TSS window

Alternatively, **users can define any window they wish in any reference genome** without using the "calc_wind.R" script as long as they provide a file in the same format described above.

3.2 Step 2: Mapping SVs to TSS windows

Second step of HYENA involves mapping SV breakpoints to the TSS windows created for each gene using the "mapsv.R" script. For this analysis, breakpoints are considered individually and as long as one of the two breakpoints that define an SV falls within a TSS window, that SV is considered to have the potential to influence the gene expression. However, if both of the breakpoints of an SV fall withina gene body (i.e. intragenic SVs), these SVs are left out of the analysis as they are unlikely to change promoter-enhancer interactions for that gene.

Arg	Description
-r, --ref	Gene annotation with TSS window
-i, --id	Sample id file
-b, --bedpe	Folder for bedpe files
--ingene	Remove SVs located entirely in a gene (default: 1)
--del	Remove small deletions (default: 0)

--dels	Small deletion size cutoff (default 10000)
--dup	Remove small duplications (default: 1)
--dups	Small duplications size cutoff (default: 10000)
--inv	Remove small inversion (default: 0)
--invs	Small inversion size cutoff (default: 10000)
-w, --write	Folder for intermediate files (default: "./intermediate/")
-x, --prefix	Prefix for output files (default: "DAT")
-v, --verbose	Verbose mode for troubleshooting (default: FALSE)

Table 2: Description of arguments for mapsv.R

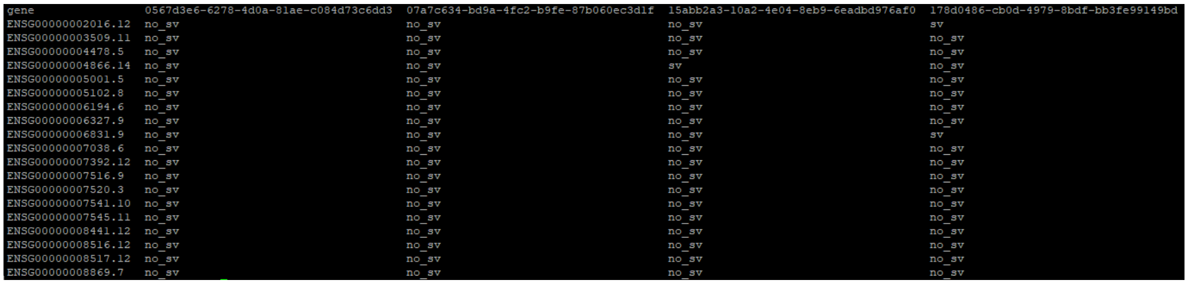
Small tandem duplications typically will not lead to new promoter-enhancer interactions. By default, they will be filtered out. However, small deletions may delete TAD boundaries or other repressive elements and new promoter-enhancer interactions can form. So small deletions are not filtered by default. However, certain tumor types have excessive number of small deletions, such as breast cancer. In this case, filter out small deletions may be beneficial.

To execute Step 2 of HYENA for THCA example, run:

```
Rscript ../R/mapsv.R \
-r ../ref/toy_gene_annot_hg19_tsswindow.txt \
-i ./data/sample_ids.txt \
-b ./data/bedpe/ \
-w ./intermediate/
```

By default, SVs located entirely in gene body are removed. Small tandem duplications are also removed, but not small deletions or inversion.

The file "sv_mapped_filtered_numSV.txt" with gene level SV genotype will be used for future steps. The file format is gene by sample ID (Figure 9). If a sample has a breakpoint within the TSS window of a gene, the sample is annotated as "sv" for that particular gene. Otherwise, the sample receives a "no_sv" status. The last two columns titled "num.sv" and "num.nosv" summarize the number of samples that have or do not have SVs, respectively, for a particular gene.



```
gene                0567d3e6-6278-4d0a-81ae-c084d73c6dd3  07a7c634-bd9a-4fc2-b9fe-87b060ec3d1f  15abb2a3-10a2-4e04-8eb9-6eadbd976af0  178d0486-cb0d-4979-8bdf-bb3fe99149bd
ENSG00000002016.12  no_sv                                 no_sv                                 no_sv                                 sv
ENSG00000003509.11  no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000004478.5   no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000004866.14  no_sv                                 no_sv                                 sv                                    no_sv
ENSG00000005001.5   no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000005102.8   no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000006194.6   no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000006327.9   no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000006831.9   no_sv                                 no_sv                                 no_sv                                 sv
ENSG00000007038.6   no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000007392.12  no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000007516.9   no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000007520.3   no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000007541.10  no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000007545.11  no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000008441.12  no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000008516.12  no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000008517.12  no_sv                                 no_sv                                 no_sv                                 no_sv
ENSG00000008869.7   no_sv                                 no_sv                                 no_sv                                 no_sv
```

Figure 9: "sv_mapped_filtered.txt" output file

When the sample size is large or the number of SVs is large, this step can be very slow. In this case, users can split the gene annotation file into smaller chunks (e.g. 1000 genes per file), run "mapsv.R" to generate multiple "sv_mapped_filtered.txt" files, and merge them into one file.

3.3 Step 3: Calculating normal scores for gene expression

If gene expression data are not normalized, they will be quantile normalized using the "quantNorm.R" script. If they are already normalized, this step can be skipped.

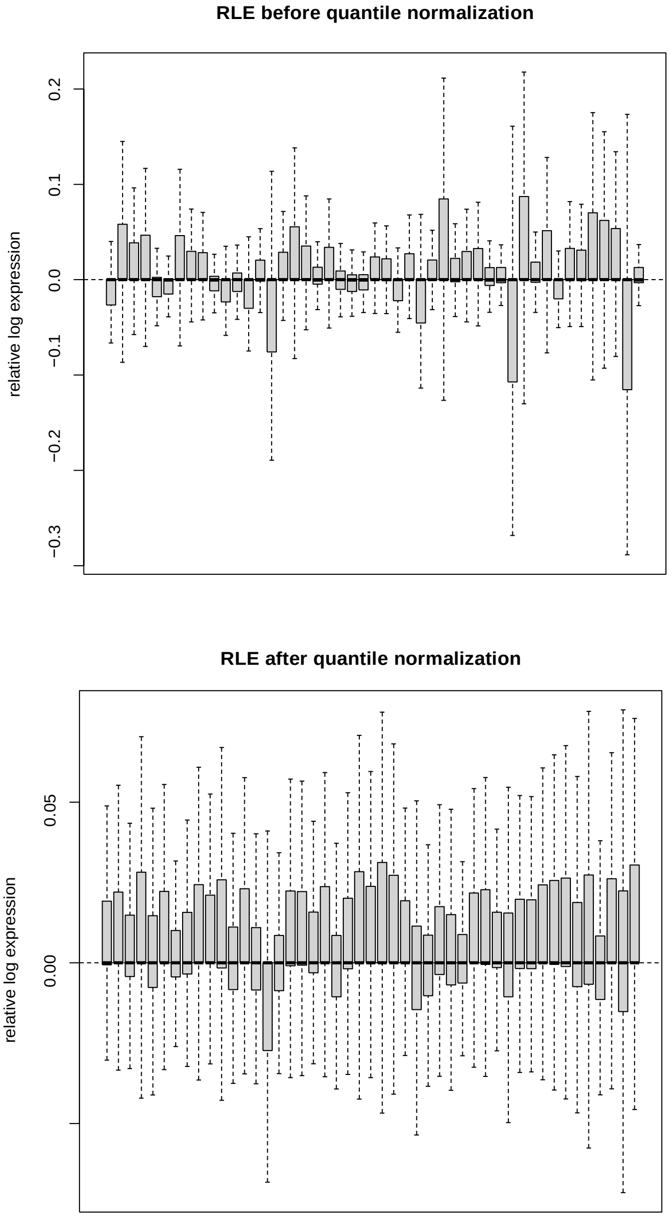
Arg	Description
-e, --exp	Gene expression input
-w, --write	Folder for intermediate files (default: "./intermediate/")
-x, --prefix	Prefix for output files (default: "DAT")
-i, --id	Sample id file

Table 3: Description of arguments for quantNorm.R and add_noise.R

Example use of quantNorm.R:

```
Rscript ../R/quantNorm.R \
-e ./data/exp.txt \
-w ./intermediate/ \
-w ./data/sample_ids.txt
```

This step generates a gene by sample ID matrix called “*.exp_quant.txt” and two relative log expression (RLE) plots showing the distribution of FPKM values before and after quantile normalization (Figure 10). These plots can be used to identify problematic or outlier samples that can skew downstream analysis.



RLE before quantile normalization

RLE after quantile normalization

Figure 10: Relative log expression across all samples before and after quantile normalization

The script “normal_score.R” will convert gene expression to normal scores. The input gene expression file is the output file of “quantNorm.R”. Then principal component analysis (PCA) is performed on gene expression scores. For THCA example, run:

```
Rscript ../R/normal_score.R \
-e ./intermediate/DAT.exp_quant.txt \
-w ./intermediate/
```

An R object file “*.exp.Rdata” will be generated and is the input gene expression file for “HYENA.R” and “pmt_exp.R”.

3.4 Step 4: Normal-score regression

“HYENA.R” tests gene expression associated with SV status by normal-score regression while correcting for the confounding factors such as gene copy number, sample purity, donor age and sex, as well as unknown factors captured by the first n principal components (PCs) of gene expression. Table 4 describes the options for "HYENA.R".

$$Expression\_norma\_\ score \sim sv\_status + CN + purity + age + sex + PC_1 + PC_2 + ... + PC_n$$

Arg	Description
-e, --exp	Expression Rdata file generated by normal_score.R or pmt_exp.R
-s, --sv	*sv_mapped_filtered_numSV.txt file generated by mapsv.R
-i, --id	Sample id file
-a, --annot	Gene annotation with TSS window
-p, --purity	Sample purity file
-c, --cna	Gene level copy number file
-m, --clinical	Clinical file
-n, --npc	Number of first n PCs to be included in the model (default: 0)
-d, --dir	Output folder (default: "./results/")
-w, --write	Folder for intermediate files (default: "./intermediate/)
-x, --prefix	Prefix for output files (default: "DAT")
-v, --verbose	Verbose mode for troubleshooting (default: FALSE)
-t, --pmt	Run in permutation mode (default: FALSE)
--pur	Include purity in regression (default: FALSE)
--cn	Include gene copy number in regression (default: FALSE)
--age	Include donor age in regression (default: FALSE)
--sex	Include donor sex in regression (default: FALSE)
--type	Include cancer type in regression for pancancer analysis (default: FALSE)
--PC	Include principal components in regression (default: FALSE)
-f, --fcutoff	SV Frequency (%) cutoff (default: 5)
-C, --cncutoff	Maximum copy number cutoff (default: 10)
-g, --genes	A list of genes of interest to be tested (default: FALSE)

Table 4: Description of arguments for HYENA.R

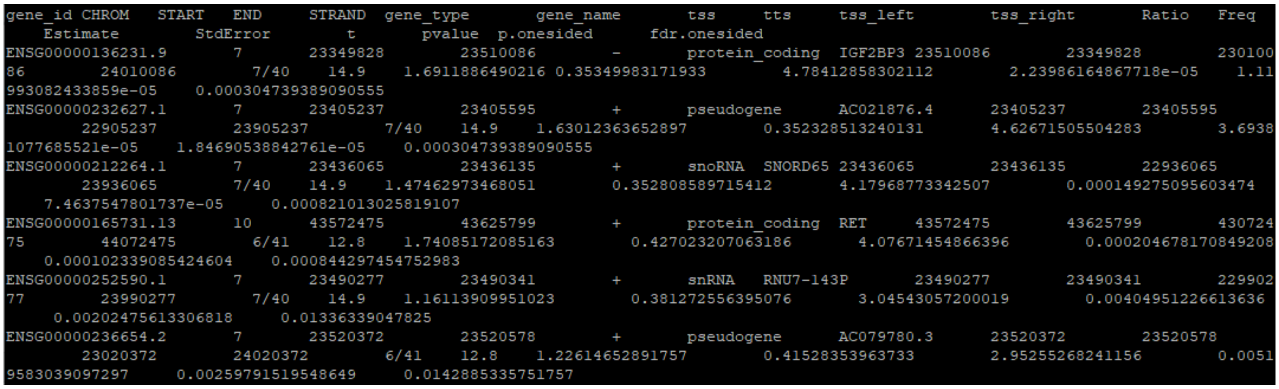
While selecting the first n PCs, HYENA tests the correlation between each PC and sv_status to make sure that those PCs that are highly correlated with sv_status for a particular gene are not included in the regression model. For example, if we wanted to include 5 PCs in the model for Gene A but the 3rd and the 5th PCs are correlated with sv_status, HYENA would include PCs 1,2,4,6, and 7 in the model. **We recommend testing number of PCs up to 10% of the sample size for small cohort.** For example, in THCA example, we test 0 to 5 PCs. **For large cohort, we recommend testing up to 20 PCs, but not more than that.**

$$Exp\_normal\_score_A \sim sv\_status_A + CN_A + purity + age + sex + PC_1 + PC_2 + PC_4 + PC_6 + PC_7$$

The users can run the full model with 0 principal component with the following command:

```
Rscript ../R/HYENA.R \
-e ./intermediate/DAT.exp.Rdata \
-s ./intermediate/DAT.sv_mapped_filtered_numSV.txt \
-i ./data/sample_ids.txt
-a ../ref/toy_gene_annot_hg19_tsswindow.txt
-p ./data/purity.txt
-c ./data/cna.txt
-m ./data/clindat.txt
-d ./results/run_PC0/
-w ./intermediate/
--pur --cn -C 10 --age --sex -f 5 --PC -n 0
```

This will generate multiple output files in the "./results/run_PC0/" directory. The most important output file is "*annot_* posEstimate.txt" with genes ranked by their false discovery rates (FDRs) (Figure 11).



```
gene_id CHROM   START   END     STRAND  gene_type       gene_name       tss     tts     tss_left        tss_right       Ratio   Freq
    Estimate        StdError        t       pvalue  p.onesided      fdr.onesided
ENSG00000136231.9       7       23349828        23510086        -       protein_coding  IGF2BP3 23510086        23349828        230100
86      24010086        7/40    14.9    1.6911886490216 0.35349983171933        4.78412858302112        2.23986164867718e-05    1.11
993082433859e-05    0.000304739389090555
ENSG00000232627.1       7       23405237        23405595        +       pseudogene      AC021876.4      23405237        23405595
        22905237        23905237        7/40    14.9    1.63012363652897        0.352328513240131       4.62671505504283        3.6938
1077685521e-05    1.84690538842761e-05    0.000304739389090555
ENSG00000212264.1       7       23436065        23436135        +       snoRNA  SNORD65 23436065        23436135        22936065
        23936065        7/40    14.9    1.47462973468051        0.352808589715412       4.17968773342507        0.000149275095603474
    7.4637547801737e-05     0.000821013025819107
ENSG00000165731.13      10      43572475        43625799        +       protein_coding  RET     43572475        43625799        430724
75      44072475        6/41    12.8    1.74085172085163        0.427023207063186       4.07671454866396        0.000204678170849208
    0.000102339085424604    0.000844297454752983
ENSG00000252590.1       7       23490277        23490341        +       snRNA   RNU7-143P       23490277        23490341        229902
77      23990277        7/40    14.9    1.16113909951023        0.381272556395076       3.04543057200019        0.00404951226613636
     0.00202475613306818     0.01336339047825
ENSG00000236654.2       7       23520372        23520578        +       pseudogene      AC079780.3      23520372        23520578
        23020372        24020372        6/41    12.8    1.22614652891757        0.41528353963733        2.95255268241156        0.0051
9583039097297     0.00259791519548649     0.0142885335751757
```

Figure 11: HYENA output

HYENA has a default cutoff of 10 for the gene's copy number. This is because genes with 10 or more copies are likely amplified in a circular extrachromosomal DNA (also known as double minutes). By default, HYENA removes these cases from downstream analysis. However, if the user would like to include these cases, they can set the "-C" argument to a very high number (e.g. 1000).

If the user has copy ratio data instead of integer copy numbers for genes, they can stilluse the "-c,--can" argument with the full path to their copy ratio file. But the format of their file shouldstill match the CN file. Moreover, if copy ratio is used, we recommend that the user removes "purity" from the regression model as the sample purity information is captured by the copy ratio data already. **For copy ratio, we recommend using a cutoff of 3.**

```
Rscript ../R/HYENA.R \
-e ./intermediate/DAT.exp.Rdata \
-s ./intermediate/DAT.sv_mapped_filtered_numSV.txt \
-i ./data/sample_ids.txt
-a ../ref/toy_gene_annot_hg19_tsswindow.txt
-c ./data/cn_ratio.txt \
-m ./data/clindat.txt
-d ./results/
-w ./intermediate/
--cn -C 3 --age --sex -f 5 --PC -n 0
```

Users can also automate testing different numbers of PCs to the regression model with the following code, which runs HYENA with 0 to 5 PCs.

```
for i in {0..5}
   do
   Rscript ../R/HYENA.R \
   -e ./intermediate/DAT.exp.Rdata \
   -s ./intermediate/DAT.sv_mapped_filtered_numSV.txt \
   -i ./data/sample_ids.txt
   -a ../ref/toy_gene_annot_hg19_tsswindow.txt
   -p ./data/purity.txt
   -c ./data/cna.txt
   -m ./data/clindat.txt
   -d ./results/run_PC${i}/
   -w ./intermediate/run_PC${i}/
   --pur --cn -C 10 --age --sex -f 5 --PC -n ${i}
   done
```

If users are running pancancer analyses and want to correct for cancer type specific effects, HYENA has the "--type" argument to include cancer type as a random variable in the normal-score regression. In this case, users should provide an extra column of "cancer_type" in the clinical data file. The following code is an example:

```
Rscript ../R/HYENA.R \
-e ./intermediate/DAT.exp.Rdata \
-s ./intermediate/DAT.sv_mapped_filtered_numSV.txt \
-i ./data/sample_ids.txt
-a ../ref/toy_gene_annot_hg19_tsswindow.txt
-p ./data/purity.txt
-c ./data/cna.txt
-m ./data/clindat.txt
-d ./results/
-w ./intermediate/
--pur --cn -C 10 --age --sex --type -f 5 --PC -n 0
```

3.5 Step 5: Calculating empirical p-values

This step is strongly recommended since p-values from the previous step are often inflated. Users can generate a null distribution by randomizing gene expression data using the "pmt_exp.R" script. Empirical p-values can be generated by repeating this permutation process multiple times. To calculate empirical p-values at the 10^{-6} level, the user needs to generate at least 1 million permuted p-values. This can be achieved by pooling p-values from multiple permutations. Note that every run of HYENA will generate approximately the number of p-values that is equal to the number of genes that have SVs satisfying the recurrence cutoff in the data set. For example, if a data set has 1,000 genes that have SVs in at least 5% of the tumors, then the user needs to run 1,000 permutations to reach final p-value count of 1 million. **We recommend running 1000 permutations for each dataset.** For tumor types with more than 10,000 genes to test, we recommend 100 permutations to reduce run time. The number of genes tested can be found in "./results/run_PC0/prefix.annot_*_PC0_posEstimate.txt" or as number of files in "./intermediate/run_PC0/matrix/" folder. **Normal-score regression on permuted data should be run in parallel.**

"pmt_exp.R" takes "*.exp.Rdata" generated by "normal_score.R" as input and shuffles sample ID at random. The output is a "*exp.Rdata" file to be used as input for "HYENA.R". If HYENA.R is being executed on permuted expression data, "--pmt" argument should be turned on.

Arg	Description
-e, --exp	Expression Rdata file generated by normal_score.R
-w, --write	Folder for intermediate files (default: "./intermediate/exp_pmt/")
-x, --prefix	Prefix for output files (default: "PMT")

Table 5: Description of arguments for pmt_exp.R

```
Rscript ../R/pmt_exp.R \
-e ./intermediate/DAT.exp.Rdata \
-w ./intermediate/exp_pmt/ \
-x PMT

Rscript ../R/HYENA.R \
-e ./intermediate/exp_PMT/PMT.exp.Rdata \
-s ./intermediate/DAT.sv_mapped_filtered_numSV.txt \
-i ./data/sample_ids.txt
-a ../ref/toy_gene_annot_hg19_tsswindow.txt
-c ./data/cna.txt \
-m ./data/clindat.txt
-d ./results/
-w ./intermediate/
--cn -C 10 --age --sex -f 5 --PC -n 0 --pmt
```

For demonstration purpose in the example pipeline, we will perform permutation only 5 times on each PC to calculate empirical p-values. The following code can automate this process in the command line:

```
# Permute expression data
for j in {1..5}
   do
   Rscript ../R/pmt_exp.R \
   -e ./intermediate/DAT.exp.Rdata \
   -w ./intermediate/exp_pmt/ \
   -x PMT${j}
   done

# Run HYENA on permuted expression
for i in {0..5}
   do
   for j in {1..5}
      do
      Rscript ../R/HYENA.R \
      -e ./intermediate/exp_pmt/PMT${j}.exp.Rdata \
      -s ./intermediate/DAT.sv_mapped_filtered_numSV.txt \
      -i ./data/sample_ids.txt \
      -a ../ref/toy_gene_annot_hg19_tsswindow.txt \
      -p ./data/purity.txt \
      -c ./data/cna.txt \
      -m ./data/clindat.txt \
      -w ./intermediate/run_PC${i}/ \
      -d ./intermediate/run_PC${i}/ \
      -x PMT${j} \
      –pur –cn -C 10 –age –sex -f 5 –PC -n ${i} --pmt
      done
   done
```

This will generate 5 permuted result files per PC run in the "./intermediate/" directory. Then the permuted p-values from these runs can be extracted and concatenated into a single file using the following command:

```
ls -d intermediate/run_PC* | while read DIR
   do
   [ -e $DIR/pvalpmt.txt ] && rm $DIR/pvalpmt.txt
   touch $DIR/pvalpmt.txt
   tail -q -n +2 $DIR/*_posEstimate.txt >> $DIR/pvalpmt.txt
   cp $DIR/pvalpmt.txt $DIR/temp.txt
   cut -f17 $DIR/temp.txt > $DIR/pvalpmt.txt
   rm $DIR/temp.txt
   done
```

Then the "pvalpmt.txt" file can be used as the null distribution to calculate empirical p-values by the "empiricalp.R" script which outputs a file with the "_adj.txt" suffix.

Arg	Description
-o, --obs	HYENA.R output file using observed gene expression
-p, --pmt	P value null distribution file (pvalpmt.txt)
-d, --dir	Output folder (default: "./results/")
--qq	Draw a QQ plot (default: FALSE)

Table 6: Description of arguments for empiricalp.R

Example use of empiricalp.R:

```
Rscript ../R/empiricalp.R \
-o ./results/*_posEstimate.txt \
-p ./intermediate/pvalpmt.txt \
-d ./results/
```

To calculate empirical p-values for each PC run:

```
for i in {0..5}
   do
   Rscript ../R/empiricalp.R \
   -o ./results/run_PC${i}/*_posEstimate.txt \
   -r ./intermediate/run_PC${i}/pvalpmt.txt \
   -d ./results/run_PC${i}/
   done
```

Users can also choose to generate a Quantile-Quantile plot (Q-Q plot) of p-values by adding the "--qq" argument. Note that the Q-Q plot argument requires all p-values to be non-zero. If any of the empirical values are 0, the Q-Q plot will not be drawn, and an error message will appear. However, the empirical p-value results will still be output appropriately.

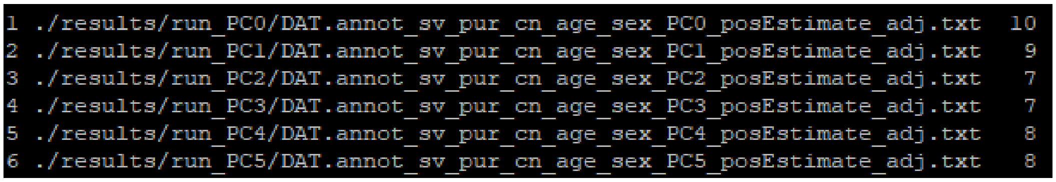
After running multiple models with different numbers of PCs, users can compare the results by “setpc.R” to identify the number of significant hits achieved by each model. A final model that identifies at least 80% of the maximum number of hits across all models with the smallest number of PCs will be picked.

Arg	Description
-l, --filelist	Comma separated list of result files
-p, --power	Desired power (default: 0.8)
-f, --fdrcutoff	FDR cut-off for significant genes (default: 0.1)
-w, --write	Folder for intermediate files (default: "./intermediate/")
--emp	P values to process are empirical p-values
-d, --dir	Output folder (default: "./results/")
-x, --prefix	Prefix for output files (default: "DAT")

Table 7: Description of arguments for setpc.R

It is crucial that the users list their results files from different models in an INCREASING order of parameters used in the model. For example, a model that uses "sv + purity + cn + age" has fewer parameters than "sv + purity + cn + age + 3 PCs". Similarly, "sv + purity + cn +age + 3 PCs" has fewer parameters than "sv + purity + cn + age + 10 PCs". This allows the script to pick the model that reaches 80% power with the fewest number of parameters to avoid over-fitting. “setpc.R” generates a result file with the extension "_sig.txt" that only has the significant hits. It will also generate a file called "Prefix.pc_tally.txt" with the number of significant genes identified in each model listed in the order provided by the user (Figure 12). An example command to run "setpc.R" would be:

```
Rscript ../R/setpc.R \
-l ./results/run_PC0/DAT.annot_sv+pur+cn+age+sex+PC0_posEstimate_adj.txt,
./results/run_PC1/DAT.annot_sv_pur_cn_age_sex_PC1_posEstimate_adj.txt,
./results/run_PC2/DAT.annot_sv_pur_cn_age_sex_PC2_posEstimate_adj.txt,
./results/run_PC3/DAT.annot_sv_pur_cn_age_sex_PC3_posEstimate_adj.txt,
./results/run_PC4/DAT.annot_sv_pur_cn_age_sex_PC4_posEstimate_adj.txt,
./results/run_PC5/DAT.annot_sv_pur_cn_age_sex_PC5_posEstimate_adj.txt \
-w ./intermediate/ \
-d ./results/ \
--emp
```



```
1 ./results/run_PC0/DAT.annot_sv_pur_cn_age_sex_PC0_posEstimate_adj.txt  10
2 ./results/run_PC1/DAT.annot_sv_pur_cn_age_sex_PC1_posEstimate_adj.txt   9
3 ./results/run_PC2/DAT.annot_sv_pur_cn_age_sex_PC2_posEstimate_adj.txt   7
4 ./results/run_PC3/DAT.annot_sv_pur_cn_age_sex_PC3_posEstimate_adj.txt   7
5 ./results/run_PC4/DAT.annot_sv_pur_cn_age_sex_PC4_posEstimate_adj.txt   8
6 ./results/run_PC5/DAT.annot_sv_pur_cn_age_sex_PC5_posEstimate_adj.txt   8
```

Figure 12: Example "pc_tally.txt" output

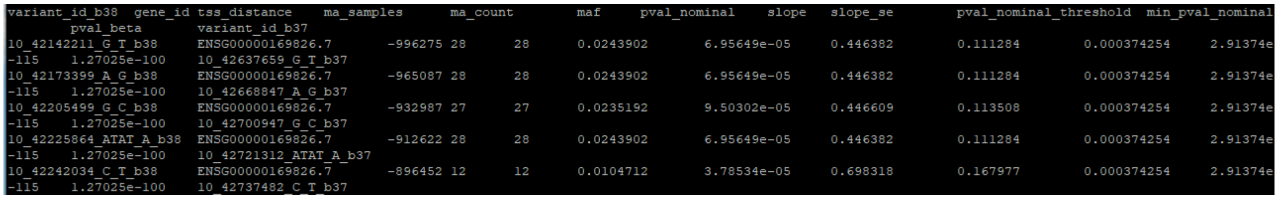
3.6 Step 6: Testing associations between SV status and known eQTLs

This step requires germline single nucleotide variations (SNVs) and is optional. Based on our experience, about 10% of the candidate genes whose expression can be partially explained by known eQTLs. “eqtl.R” can test associations between SV status and known eQTLs and remove genes with significant associations.

Arg	Description
-Q, --eqtl	Tissue-specific significant gene-eQTL pairs
-S, --snv	Germline SNV genotypes for all samples
-R, --results	HYENA results file '_posEstimate.txt' or '_adj.txt' or'_adj_sig.txt'
--emp	P values are empirical p-values
-f, --fdrcutoff	FDR cutoff for genes (default: 0.1)
-c, --sigcutoff	P-value cutoff for eQTLs (default: 0.05)
-g, --genome	Reference genome version ('hg19' or 'hg38')
-x, --prefix	Prefix for output files (default: "DAT")
-V, --svstat	*sv_mapped_filtered_numSV.txt generated by mapsv.R in intermediate folder
-i, --id	Sample id file
-a, --annot	Gene annotation with TSS window
-w, --write	Folder for intermediate files (default "./intermediate/ ")
-d, --dir	Output folder (default "./results/")
-v, --verbose	Verbose mode for troubleshooting (default: FALSE)

Table 8: Description of arguments for eqtl.R

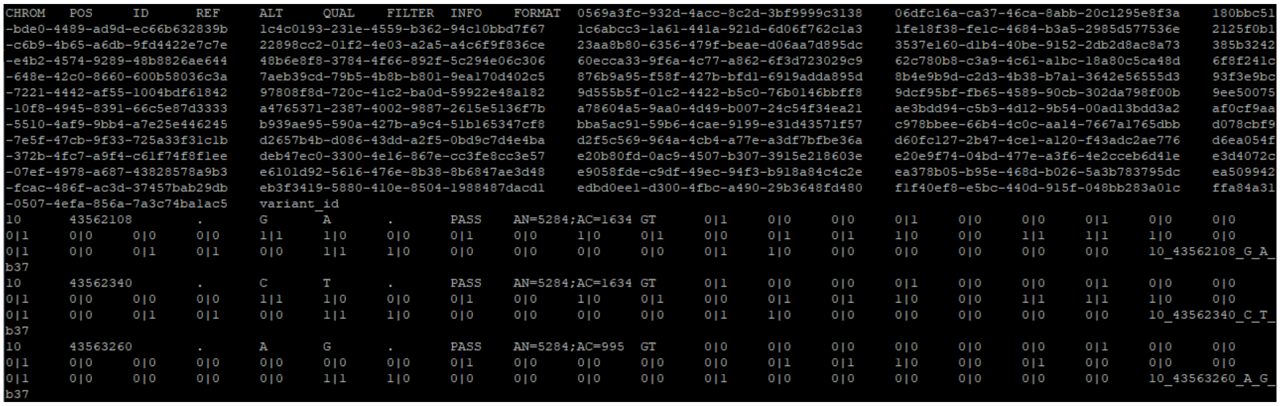
“eqtl.R” script requires two SNV and eQTL input files: (1) germline SNV genotypes called from normal samples and (2) known eQTL-gene pairs for the relevant tissue. eQTL-gene pairs can be downloaded from the Genotype-Tissue Expression (GTEx) data portal (http://www.gtexportal.org). Each row should indicate one unique eQTL with an ID in the format of “CHROM_POS_REF_ALT_version” (Figure 13. The “variant_id” should end with “_b38” if hg38 is used as the reference genome, and “_b37” if hg19 is used. GTEx portal provides liftover references to convert variant ids between hg38 and hg19. If the user provides two columns for variant id (one for hg19 id and one for hg38 id in the GTEx reference file as in Figure 13), “eqtl.R” will automatically select the appropriate id column based on the reference version indicated in the “--genome” argument.



```
variant_id_b38  gene_id tss_distance    ma_samples      ma_count        maf     pval_nominal    slope   slope_se        pval_nominal_threshold  min_pval_nominal
        pval_beta       variant_id_b37
10_42142211_G_T_b38     ENSG00000169826.7       -996275 28      28      0.0243902       6.95649e-05     0.446382        0.111284        0.000374254     2.91374e
-115    1.27025e-100    10_42637659_G_T_b37
10_42173399_A_G_b38     ENSG00000169826.7       -965087 28      28      0.0243902       6.95649e-05     0.446382        0.111284        0.000374254     2.91374e
-115    1.27025e-100    10_42668847_A_G_b37
10_42205499_G_C_b38     ENSG00000169826.7       -932987 27      27      0.0235192       9.50302e-05     0.446609        0.113508        0.000374254     2.91374e
-115    1.27025e-100    10_42700947_G_C_b37
10_42225864_ATAT_A_b38  ENSG00000169826.7       -912622 28      28      0.0243902       6.95649e-05     0.446382        0.111284        0.000374254     2.91374e
-115    1.27025e-100    10_42721312_ATAT_A_b37
10_42242034_C_T_b38     ENSG00000169826.7       -896452 12      12      0.0104712       3.78534e-05     0.698318        0.167977        0.000374254     2.91374e
-115    1.27025e-100    10_42737482_C_T_b37
```

Figure 13: "Thyroid.signifpairs.goi.txt" GTEx eQTL-gene pair data format

SNV data for the normal samples within the data set should be compiled in the format depicted in Figure 14. The contents of the “variant_id” column in the SNV file should match the formatting and reference genome version used in the variant id column of the GTEx file.



```
CHROM   POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  0569a3fc-932d-4acc-8c2d-3bf9999c3138    06dfc16a-ca37-46ca-8abb-20c1295e8f3a    180bbc51
-bde0-4489-ad9d-ec66b632839b    1c4c0193-231e-4559-b362-94c10bbd7f67    1c6abcc3-1a61-441a-921d-6d06f762c1a3    1fe18f38-fe1c-4684-b3a5-2985d577536e    2125f0b1
-c6b9-4b65-a6db-9fd4422e7c7e    22898cc2-01f2-4e03-a2a5-a4c6f9f836ce    23aa8b80-6356-479f-beae-d06aa7d895dc    3537e160-d1b4-40be-9152-2db2d8ac8a73    385b3242
-e4b2-4574-9289-48b8826ae644    48b6e8f8-3784-4f66-892f-5c294e06c306    60ecca33-9f6a-4c77-a862-6f3d723029c9    62c780b8-c3a9-4c61-a1bc-18a80c5ca48d    6f8f241c
-648e-42c0-8660-600b58036c3a    7aeb39cd-79b5-4b8b-b801-9ea170d402c5    876b9a95-f58f-427b-bfd1-6919adda895d    8b4e9b9d-c2d3-4b38-b7a1-3642e56555d3    93f3e9bc
-7221-4442-af55-1004bdf61842    97808f8d-720c-41c2-ba0d-59922e48a182    9d555b5f-01c2-4422-b5c0-76b0146bbff8    9dcf95bf-fb65-4589-90cb-302da798f00b    9ee50075
-10f8-4945-8391-66c5e87d3333    a4765371-2387-4002-9887-2615e5136f7b    a78604a5-9aa0-4d49-b007-24c54f34ea21    ae3bdd94-c5b3-4d12-9b54-00ad13bdd3a2    af0cf9aa
-5510-4af9-9bb4-a7e25e446245    b939ae95-590a-427b-a9c4-51b165347cf8    bba5ac91-59b6-4cae-9199-e31d43571f57    c978bbee-66b4-4c0c-aa14-7667a1765dbb    d078cbf9
-7e5f-47cb-9f33-725a33f31c1b    d2657b4b-d086-43dd-a2f5-0bd9c7d4e4ba    d2f5c569-964a-4cb4-a77e-a3df7bfbe36a    d60fc127-2b47-4ce1-a120-f43adc2ae776    d6ea054f
-372b-4fc7-a9f4-c61f74f8f1ee    deb47ec0-3300-4e16-867e-cc3fe8cc3e57    e20b80fd-0ac9-4507-b307-3915e218603e    e20e9f74-04bd-477e-a3f6-4e2cceb6d41e    e3d4072c
-07ef-4978-a687-43828578a9b3    e6101d92-5616-476e-8b38-8b6847ae3d48    e9058fde-c9df-49ec-94f3-b918a84c4c2e    ea378b05-b95e-468d-b026-5a3b783795dc    ea509942
-fcac-486f-ac3d-37457bab29db    eb3f3419-5880-410e-8504-1988487dacd1    edbd0ee1-d300-4fbc-a490-29b3648fd480    f1f40ef8-e5bc-440d-915f-048bb283a01c    ffa84a31
-0507-4efa-856a-7a3c74ba1ac5    variant_id
10      43562108        .       G       A       .       PASS    AN=5284;AC=1634 GT      0|1     0|0     0|0     0|1     0|0     0|0     0|1     0|0     0|0
0|1     0|0     0|0     0|0     1|1     1|0     0|0     0|1     0|0     1|0     0|1     0|0     0|1     0|1     1|0     0|0     1|1     1|1     1|0     0|0
0|1     0|0     0|1     0|1     0|0     1|1     1|0     0|0     0|0     0|0     0|0     0|1     1|0     0|0     0|0     0|0     0|0     0|0     10_43562108_G_A_
b37
10      43562340        .       C       T       .       PASS    AN=5284;AC=1634 GT      0|1     0|0     0|0     0|1     0|0     0|0     0|1     0|0     0|0
0|1     0|0     0|0     0|0     1|1     1|0     0|0     0|1     0|0     1|0     0|1     0|0     0|1     0|1     1|0     0|0     1|1     1|1     1|0     0|0
0|1     0|0     0|1     0|1     0|0     1|1     1|0     0|0     0|0     0|0     0|0     0|1     1|0     0|0     0|0     0|0     0|0     0|0     10_43562340_C_T_
b37
10      43563260        .       A       G       .       PASS    AN=5284;AC=995  GT      0|0     0|0     0|0     0|0     0|0     0|0     0|1     0|0     0|0
0|1     0|0     0|0     0|0     0|0     0|0     0|0     0|1     0|0     0|0     0|0     0|0     0|1     0|1     1|0     0|0     0|1     0|0     0|0     0|0
0|1     0|0     0|0     0|0     0|0     1|1     1|0     0|0     0|0     0|0     0|0     0|1     0|0     0|0     0|0     0|0     0|0     0|0     10_43563260_A_G_
b37
```

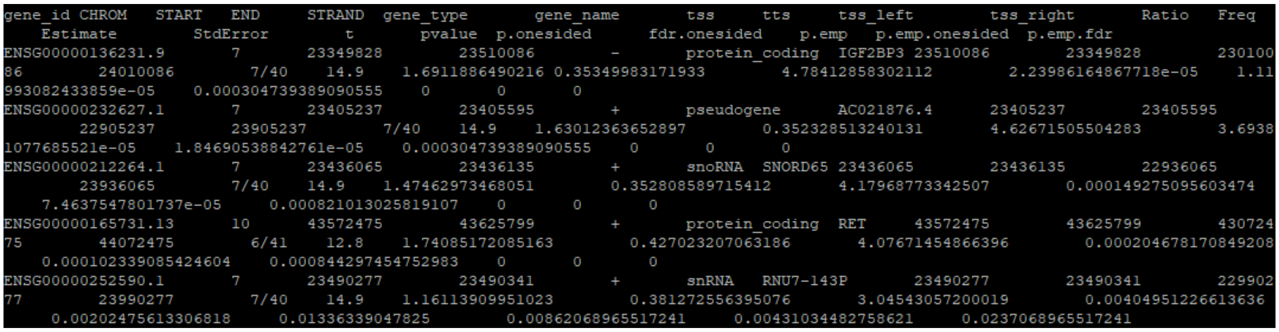
Figure 14: "snv.txt" input file format

Running the following command will filter genes based on eQTL and SV status association. If a gene has at least one eQTL that correlates with the SV genotype of the gene, that gene is removed. A candidate gene list after filtering will be generated with the extension "_sig_rmeqtl.txt". Note that if there are multiple “_posEstimate_adj_sig.txt” files in the “./results/” folder, an error will occur when running the command below. There should be only one such file.

```
Rscript ../R/eqtl.R \
-Q ./ref/Thyroid.signifpairs.goi.txt \
-S ./data/snv.txt \
-R ./results/*_posEstimate_adj_sig.txt \
-V ./intermediate/run_PC0/DAT. sv_mapped_filtered_numSV.txt \
-i ./data/sample_ids.txt \
-a ../ref/toy_gene_annot_hg19_tsswindow.txt \
-w ./intermediate/ \
-d ./results/ \
--emp
```

4 Results

The file with candidate gene list depends on how many steps users execute. If HYENA is executed without empirical p-value calculations, the output file will be labeled <Prefix>.<Parameters>_posEstimate.txt in the output folder. If empirical p-values are calculated, the output file will have the “_adj.txt” suffix. If multiple PCs are tested and “setpc.R” script is used to select regression model, then the output will have the "_sig.txt" suffix and contain only the significant genes from the selected model. Note that “setpc.R” can be used on both regression p-values and empirical p-values. If “eqtl.R” is used, the final output will have the "_rmeqtl.txt" suffix. The output file for THCA example is “DAT.annot_sv_pur_cn_age_sex_PC0_posEstimate_adj_sig_rmeqtl.txt” shown in Figure 15. Users may generate a file with different number of PC being selected.



```
gene_id CHROM   START   END     STRAND  gene_type       gene_name       tss     tts     tss_left        tss_right       Ratio   Freq
    Estimate        StdError        t       pvalue  p.onesided      fdr.onesided    p.emp   p.emp.onesided  p.emp.fdr
ENSG00000136231.9       7       23349828        23510086        -       protein_coding  IGF2BP3 23510086        23349828        230100
86      24010086        7/40    14.9    1.6911886490216 0.35349983171933        4.78412858302112        2.23986164867718e-05    1.11
993082433859e-05    0.000304739389090555    0       0       0
ENSG00000232627.1       7       23405237        23405595        +       pseudogene      AC021876.4      23405237        23405595
        22905237        23905237        7/40    14.9    1.63012363652897        0.352328513240131       4.62671505504283        3.6938
1077685521e-05    1.84690538842761e-05    0.000304739389090555    0       0       0
ENSG00000212264.1       7       23436065        23436135        +       snoRNA  SNORD65 23436065        23436135        22936065
        23936065        7/40    14.9    1.47462973468051        0.352808589715412       4.17968773342507        0.000149275095603474
    7.46375478017337e-05    0.000821013025819107    0       0       0
ENSG00000165731.13      10      43572475        43625799        +       protein_coding  RET     43572475        43625799        430724
75      44072475        6/41    12.8    1.74085172085163        0.427023207063186       4.07671454866396        0.000204678170849208
    0.000102339085424604    0.000844297454752983    0       0       0
ENSG00000252590.1       7       23490277        23490341        +       snRNA   RNU7-143P       23490277        23490341        229902
77      23990277        7/40    14.9    1.16113909951023        0.381272556395076       3.04543057200019        0.00404951226613636
    0.00202475613306818     0.01336339047825        0.00862068965517241     0.00431034482758621     0.0237068965517241
```

Figure 15: Final output file for THCA example

If the sample size is large (e.g. more than 1000), HYENA can be slow. It is recommended to split the reference gene annotation file into smaller chunks (e.g. 1000 gene per file). Each subset of genes can be run in parallel. After running HYENA.R, the result files "./results/run_PC*/prefix.annot_*_PC*_posEstimate.txt" can be combined.

5 Plotting

Users can plot gene expression for the candidate genes. The expression plots are grouped based on the SV genotype ("sv" vs. "no_sv"). If cancer type was a part of the regression model, the plot can be grouped by cancer type as well.

Arg	Description
-R, --results	HYENA results file with the significant genes '_sig.txt'
--type	Group samples by cancer type (default: FALSE)
-d, --dir	Folder for gene expression matrices generated by HYENA.R
	(default: " ./intermediate/run_PC0/matrix/")
-w, --write	Output folder (default: " ./results/plots/")

Table 9: Description of arguments for expr_plotter.R

To generate expression plots for the example data set, run the following code:

```
Rscript ../R/expr_plotter.R \
-R ./DAT.annot_sv_pur_cn_age_sex_PC0_posEstimate_adj_sig_rmeqtl.txt \
-d ./intermediate/run_PC0/matrix/ \
-w ./results/plots/
```

"expr_plotter.R" generates a file in the output folder with "_fc.txt" suffix containing expression fold change (fpkm-qn) for each gene. The fold changes provided here should be used by caution. The significant genes are nominated by normal score regression. The estimate from normal-score regression reflects changes in ranks. So, we provide fold changes in this step calculated by linear regression.

6 Reference and contact

If you have used HYENA package, please cite our publication: TBD.

If you are unable to run the THCA example, it is likely due to package dependencies. You need to ensure all the required packages are installed. Sometimes, a package can have a newer version with a slight change to its functions. That may cause errors in HYENA. It is not always possible to install an older version. You may have to identify the faulty function and modify the HYENA code.

If you are able to run the THCA example, but unable to run your own data, most likely it is an input file formatting problem. You may double check **gene ID matching**, **chromosome name matching** ("chr1" and "1"), **sample ID matching**, **special characters in sample IDs**, **field separators** ("tab"), **NA** or **missing values** in gene expression matrix, **non-numeric values** for gene expression and copy number, **truncated files**, **mis-spelled column names**, **DOS/Mac new line characters**, **empty lines** in input files, etc.

If you have questions using HYENA, please contact Lixing Yang (lixingyang@uchicago.edu).