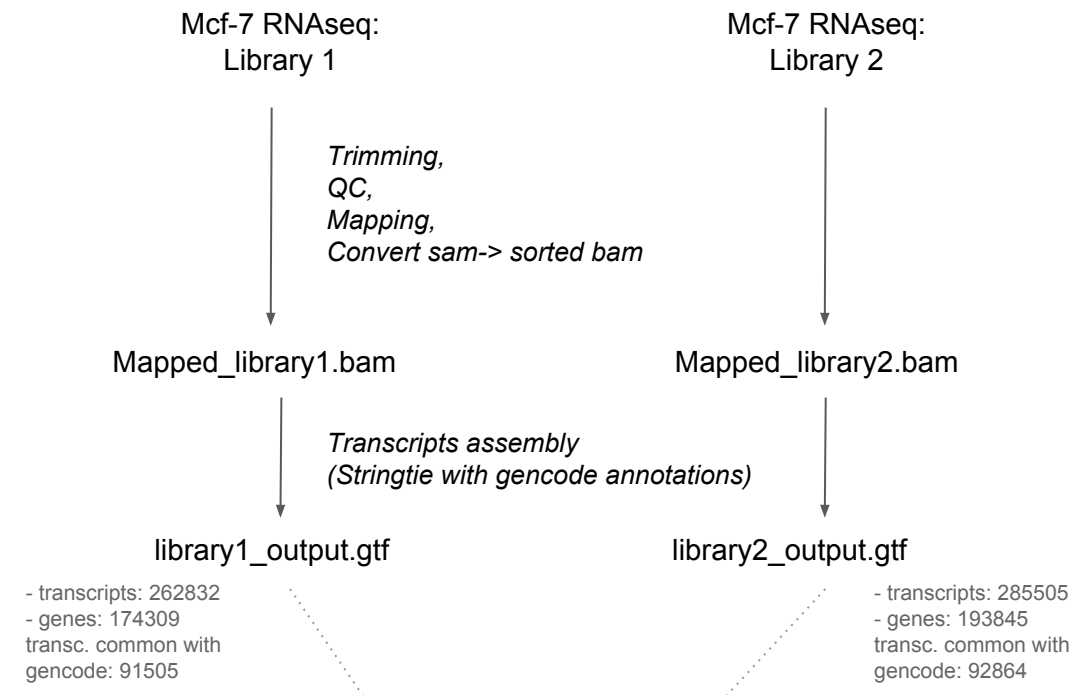
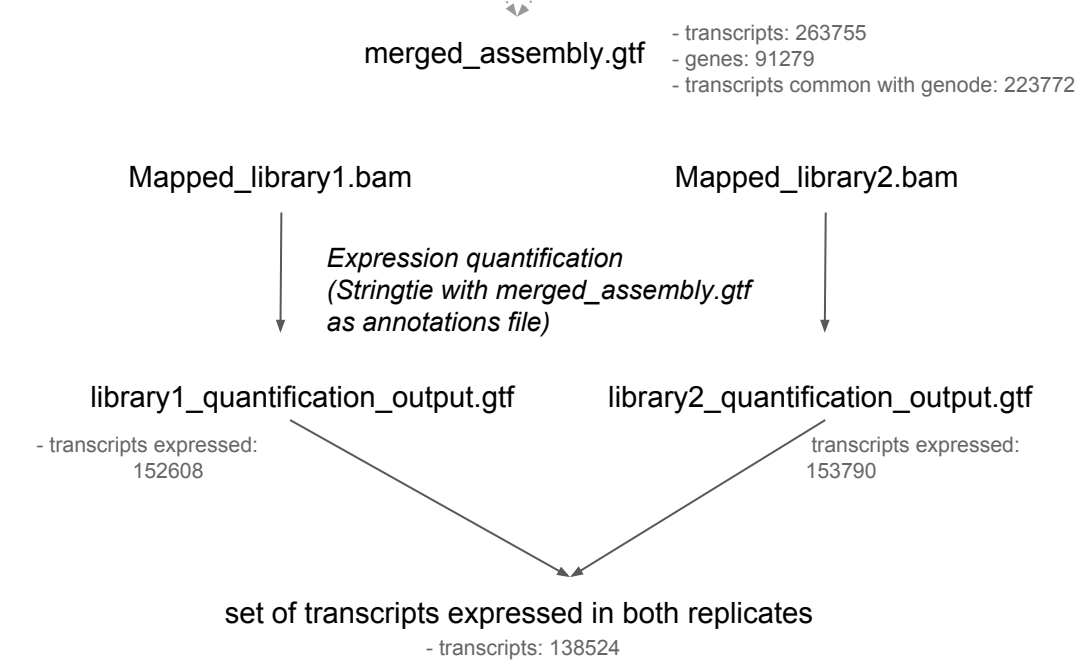
Part 1: Annotate Mcf7-expressed lincRNAs and PCGs

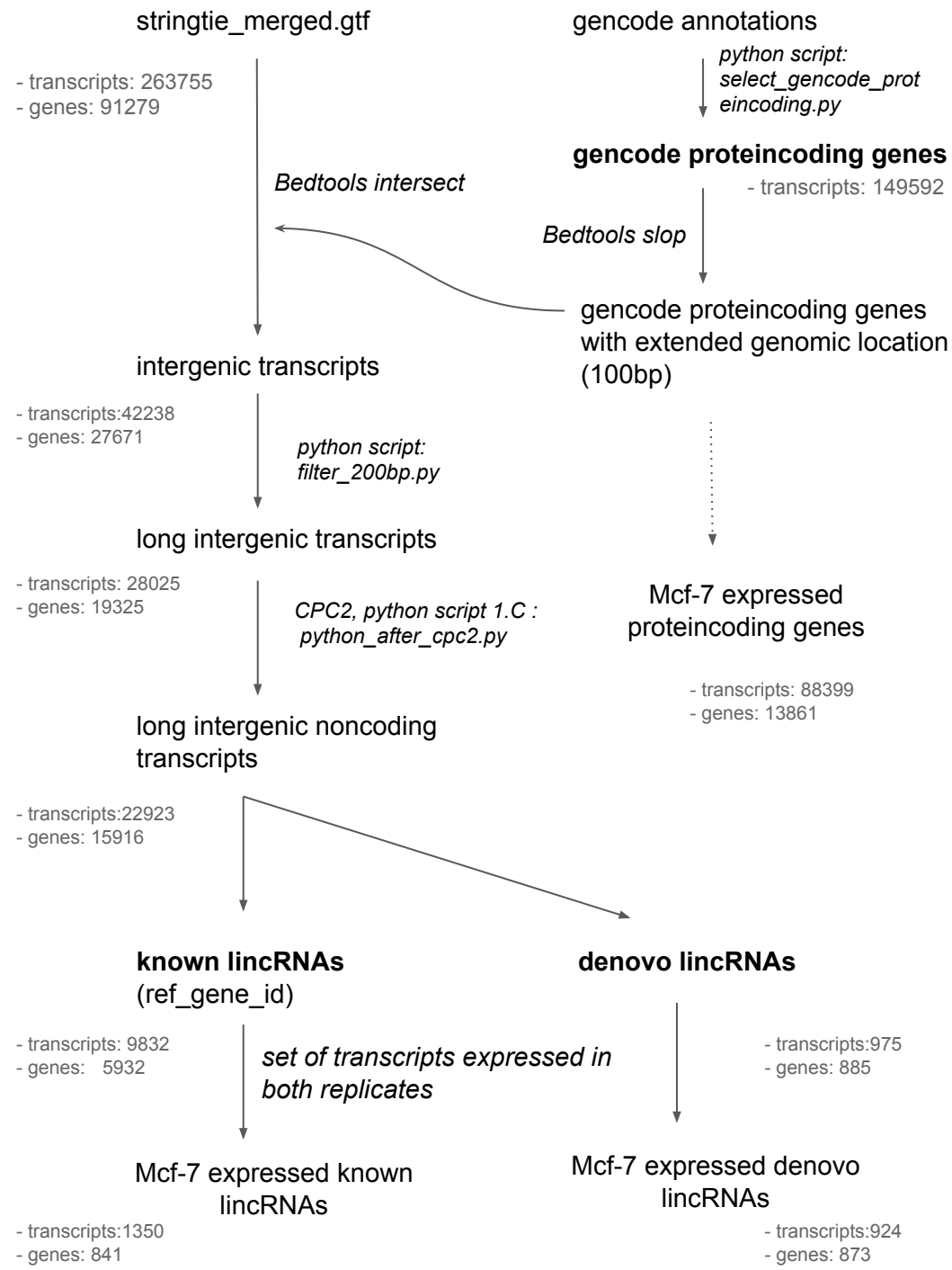
1.A. Build a set of expressed transcripts with libraries



1.B. Quantify expressed transcripts in both libraries



1.C. Identify lincRNAs and PCGs



Final .gtf

Gencode PCGs: 149592 transcripts

Known lincRNAs: 9832 transcripts

Denovo lincRNAs: 975 transcripts

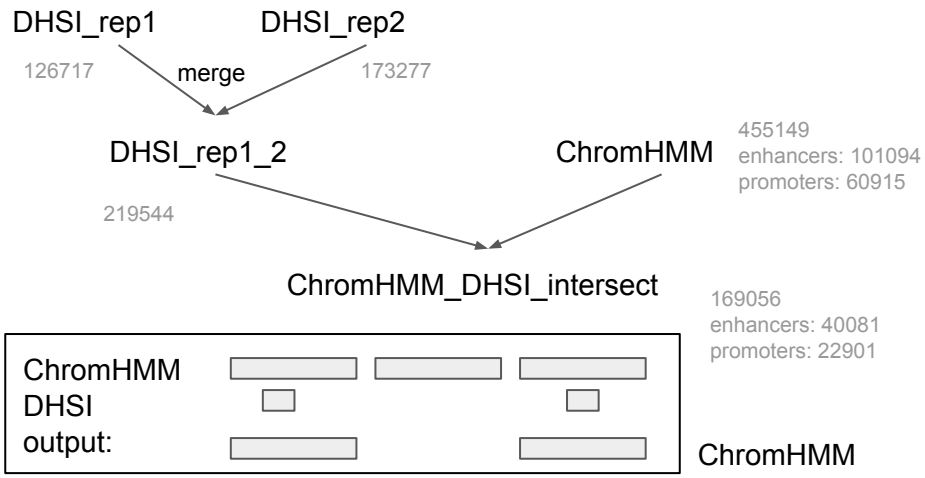
Scripts

A. Mapping Mcf-7 RNAseq raw data

1.	bash_annotations.sh	→	python_annotations.py
1.A1	bash_library1.sh	→	python_library1.py
1.A2	bash_library2.sh	→	python_library2.py
1.B	bash_merge.sh	→	python_merge.py
1.C	bash_after_cpc2.sh	→	python_after_cpc2.py

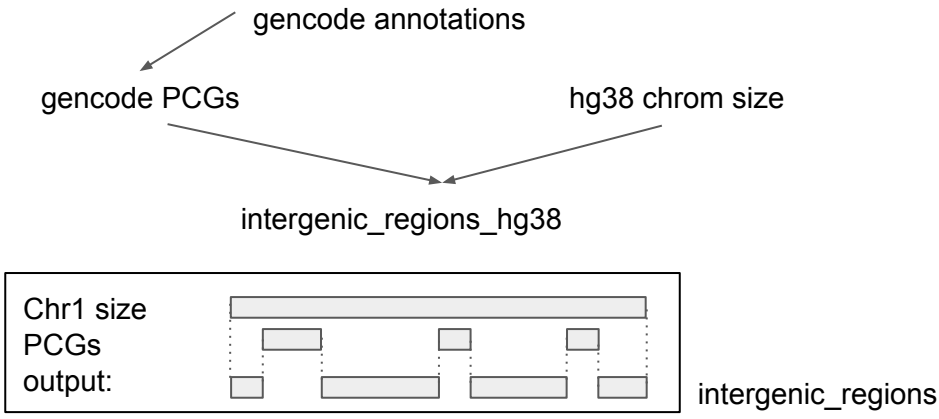
Part 2: Annotate elincRNAs and plincRNAs

2.A. Overlap enhancers with DHSI regions to get accessible enhancers



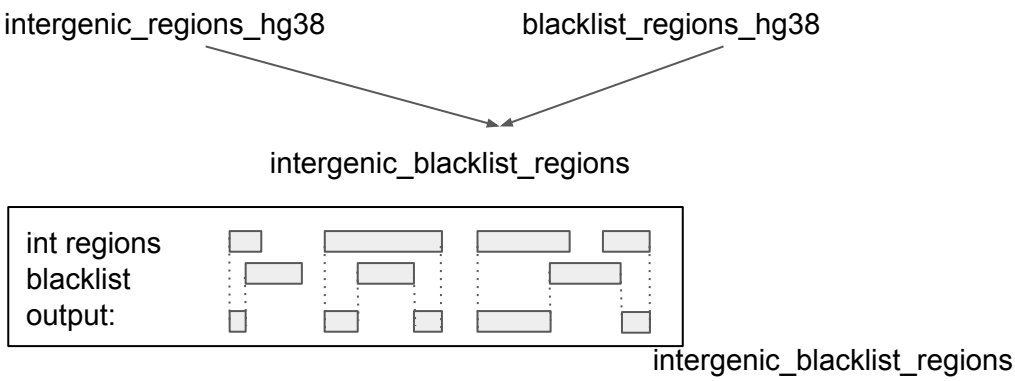
bedtools intersect -a ChromHMM_MCF7_hg38.bed -b DHSI_MCF7_rep1_2.bed -u > ChromHMM_DHSI_intersect.bed

2.B. Get intergenic regions of the genome



bedtools complement -i gencode_proteincoding_genes_only_sorted.gtf -g hg38.chrom_sorted > intergenic_regions_hg38.gtf

2.C. Exclude blacklist regions from hg38 intergenic region



~/bedops/bin/bedops --difference intergenic_regions_hg38.gtf hg38.blacklist_sorted.bed > intergenic_blacklist_region.bed

2.D. Overlap accessible chromHMM with intergenic region to get intergenic chromHMM (promoters and enhancers)

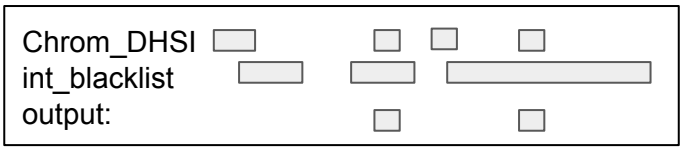
ChromHMM_DHSI_intersect.bed

169056
enhancers: 40081
promoters: 22901

intergenic_blacklist_regions

intergenic_chromHMM

70735
enhancers: 16696
promoters: 5235



intergenic_chromHMM

bedtools intersect -a ChromHMM_DHSI_intersect.bed -b intergenic_blacklist_region.bed -f 1.0 -u > intergenic_chromHMM.bed

→ accessible and fully intergenic promoters and enhancers

---

intergenic_chromHMM → intergenic_enhancers 16696
intergenic_chromHMM → intergenic_promoters 5235

ChromHMM_DHSI → enhancers 40081
ChromHMM_DHSI → promoters 22901

2.E.Get TSS from MCF7 CAGE reads (detailed pipeline: 'CAGE clustering')

hg19.cage_reads → hg38.cage_reads_rep1_rep2_rep3

hg19.cage_reads_
EGF1_00h00_rep1

hg19.cage_reads_
EGF1_00h00_rep2

*python script:
cluster_CAGE_reads_into_TSS.py*

hg38.cage_reads_rep1_rep2_rep3 → TSS_cage_cluster_v3.bed

431477

2.F. Overlap intergenic enhancers with CAGE TIRs to get enhancer-associated TIRs (eTIRs)

intergenic eTIRs: 5956 (4453 enhancers)

bedtools intersect -a TSS_cage_cluster.bed -b intergenic_enhancers.bed -u > intergenic_enhancer_associated_TIRs.bed

intergenic pTIRs: 4504 (3070 promoters)

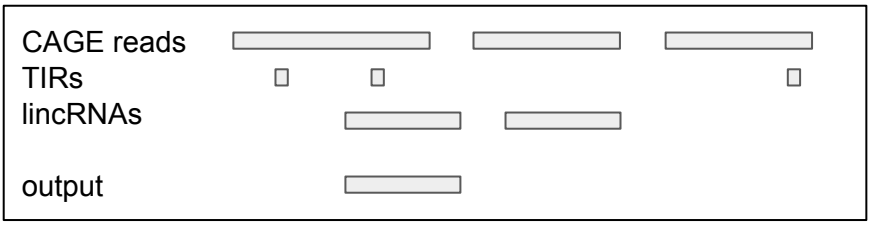
bedtools intersect -a TSS_cage_cluster.bed -b intergenic_promoters.bed -u > intergenic_promoter_associated_TIRs.bed

pTIRs: 29391

bedtools intersect -a TSS_cage_cluster.bed -b promoters.bed -u > promoter_associated_TIRs.bed

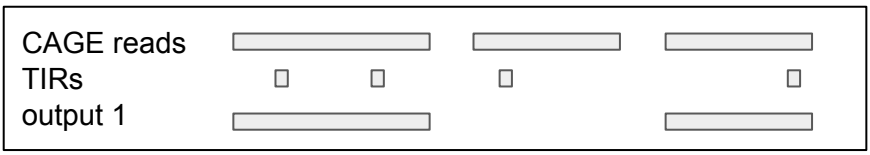
2.G. Determine expressed elincRNAs/plincRNAs/PCGs using CAGE data

Method: determine whether there is a read that overlap both a TIR and a lincRNA



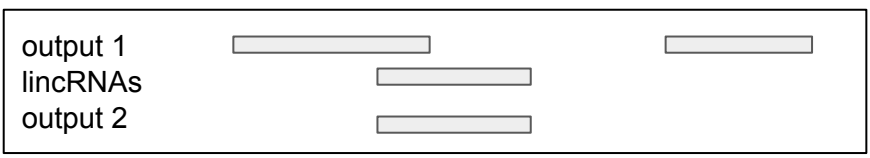
Commands written for elincRNAs. Repeted for plincRNAs and PCGs

A



bedtools intersect -a cage_reads_hg38_sorted.bed -b intergenic_enhancer_associated_TIRs.bed -s -u > cage_associated_eTIRs.bed

B



bedtools intersect -a breast_exp_lincrnas_first_exon.gtf -b cage_associated_eTIRs.bed -u -s > elincRNAs.gtf

breast_exp_lincrnas_first_exon.gtf	2274
breast_exp_PCGs_first_exon.gtf	88399

TIRs	431477
intergenic_enhancer_associated_TIRs.bed	5956
intergenic_promoter_associated_TIRs.bed	4504
promoter_associated_TIRs.bed	29391
cage_associated_eTIRs.bed	555103
cage_associated_pTIRs.bed	2082626
cage_associated_pcg_TIRs.bed	8537013
elincRNAs.gtf	104
plincRNAs.gtf	252
pPCGs.gtf	25521

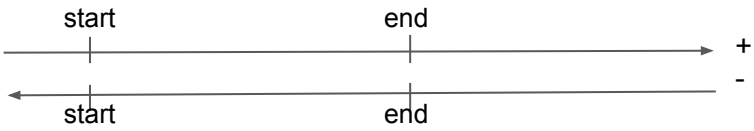
Analysis V3 transcript exon count: 1 exon / 2 exons / more than 2 exons

elincRNAs:	25 / 38 / 41	ratio: 3.16
plincRNAs:	82 / 68 / 102	ratio: 2.07
pPCG:	384 / 2452 / 22685	ratio: 65.46

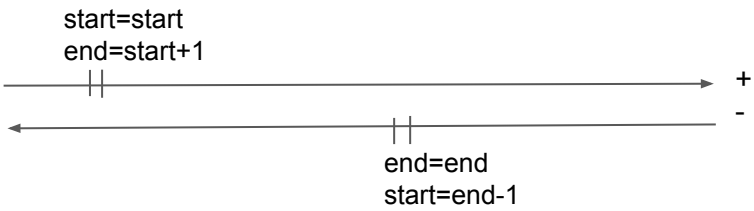
CAGE clustering

1. extract the 5′ end position of each read (read TSSs)

Original file:

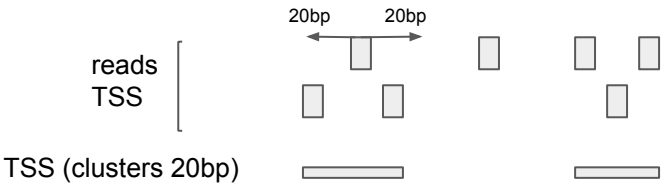


Modified file:



→ file: “cage_reads_hg38_start_start.bed”

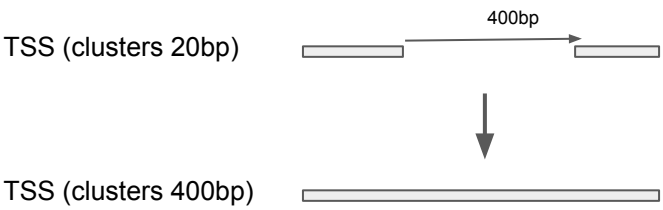
2. cluster reads TSS closer than 20bp



“bedtools merge -i cage_reads_hg38_start_start_sorted.bed -s -d 20 -c 1,6 -o count, distinct > cage_clusters_20bp.bed”

→ keep only clusters that contain 2 reads or more

3. cluster TSS closer than 400bp



“*bedtools merge -i cage_cluster_20bp_pairs_sorted.bed -s -d 400 -c 1,5,6 -o count, sum, distinct > cage_clusters_400bp.bed”*

→ keep only clusters that contain 5 reads or more
→ TSS_cage_cluster.bed

script: /scratch/beegfs/monthly/rhofmeis_elinc/CAGE/cluster_CAGE_reads_into_TSS.py

Part 3: Quantify lincRNAs and PCGs expression level in GTEx breast RNAseq

Mapping SRA scripts:

bash_script_empty.sh ⟶ python_script_empty.py

↓

Prefetch download SRA
Trimmomatic,
QC,
Hisat2 (mapping),
Stringtie (transcripts assembly, expression quantification)

make_scripts.py :

→ for each SRA:
- → replace in python_script: SRRxxx by SRA number
- → replace in bash_script: SRRxxx by SRA number
- → run bash_script → run python_script

3.A. Store results to save space

Implemented in the *python_script*, a function modifies the output (gtf format ~260M) to keep only the necessary information (~5.5M).

3.a. Transcripts expression level file for one indiv.

transcript_id	Fpkm	Tpm
ENST00000473358.1	0.000000	0.000000
ENST00000417324.1	0.000000	0.000000
ENST00000461467.1	0.000000	0.000000
ENST00000641515.2	0.000000	0.000000
ENST00000335137.4	0.000000	0.000000
ENST00000477740.5	0.000000	0.000000
ENST00000453576.2	0.000000	0.000000
ENST00000496488.1	0.000000	0.000000
MSTRG.78.1	3.591500	7.411879

3.B. Verify, debug, re-run scripts

1. Implemented in the script *make_scripts.py*, a function create a file (*done.txt*)that list all SRR to mapp and quantify.

3.b.1 Original done.txt file

SRR1068977	not_processed
SRR1068999	not_processed
SRR1070208	not_processed
SRR1070260	not_processed
SRR1070738	not_processed
SRR1071084	not_processed
SRR1071905	not_processed
SRR1074860	not_processed
SRR1075484	not_processed
SRR1076219	not_processed
SRR1076441	not_processed
SRR1077139	not_processed
SRR1077920	not_processed
SRR1078258	not_processed

2. Implemented in the script *python_script.py*, a function modifies *done.txt* to replace ‘not_processed’ by the size of the output.

3.b.2 done.txt file after the first run

SRR1068977	5711028
SRR1068999	5672836
SRR1070208	not_processed
SRR1070260	5710846
SRR1070738	5671154
SRR1071084	5690314
SRR1071905	not_processed
SRR1074860	5682464
SRR1075484	5702848
SRR1076219	5687583
SRR1076441	not_processed
SRR1077139	5709827
SRR1077920	5706540
SRR1078258	not_processed

3. script: *get_not_processed.py*
Reads the file *done.txt* and re-run script for SRR number labelled as ‘not_processed’

4. script: *get_not_processed_vdb-cache.py*
After bug identified for unprocessed jobs, this script remove bugging lines considered not important and re-run scripts.

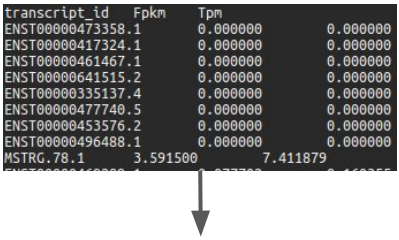
5. script: *extract_err_file_done_v2.py*
Once all jobs are completed, this script reads extract %trimmed and %mapped reads and add them to *done.txt*

3.b.2 done_v2.txt file one all jobs completed

SRA	file_size	%mapping	trimming:both surv/dropped
SRR1068977	5711028	98.63%	['86.53%/3.62%']
SRR1068999	5672836	98.60%	['86.91%/5.03%']
SRR1070208	5686865	98.56%	['88.71%/4.62%']
SRR1070260	5710846	98.34%	['87.55%/4.63%']
SRR1070738	5671154	98.71%	['87.80%/4.61%']
SRR1071084	5690314	97.99%	['90.26%/3.10%']
SRR1071905	5681685	98.60%	['92.35%/2.52%']
SRR1074860	5682464	98.67%	['89.50%/4.25%']
SRR1075484	5702848	98.08%	['89.65%/3.42%']
SRR1076219	5687583	98.58%	['87.06%/5.09%']
SRR1076441	5677204	98.66%	['89.67%/2.66%']
SRR1077139	5709827	98.38%	['88.72%/4.23%']
SRR1077920	5706540	98.03%	['91.41%/3.13%']
SRR1078258	5705450	98.42%	['88.82%/4.02%']

3.C. Build phenotype table to call cis-eQTL

script: *make_table_phenotype.py*
This script reads the 224 output file and write a phenotype table that contains as rows transcripts and as columns their chr, start, end, and expression level in each of the 224 individuals.



transcript_id	Fpkm	Tpm
ENST00000473358.1	0.000000	0.000000
ENST00000417324.1	0.000000	0.000000
ENST00000461467.1	0.000000	0.000000
ENST00000641515.2	0.000000	0.000000
ENST00000335137.4	0.000000	0.000000
ENST00000477740.5	0.000000	0.000000
ENST00000453576.2	0.000000	0.000000
ENST00000496488.1	0.000000	0.000000
MSTRG.78.1	3.591500	7.411879

x 224

#chr	start	end	transcript_id	GTEX-XBED	GTEX-U8XE	GTEX-XV7Q	GTEX-XOT4	GTEX-RUSQ
chr1	65419	71585	ENST00000641515.2	0.000000	0.000000	0.000000	0.000000	0.000000
chr1	69055	70108	ENST00000335137.4	0.075827	0.000000	0.000000	0.000000	0.000000
chr1	450703	451697	ENST00000426406.3	0.040425	0.028647	0.055988	0.021569	0.018998
chr1	685679	686673	ENST00000332831.4	0.038286	0.028647	0.055988	0.021569	0.018998
chr1	923928	939291	ENST00000420190.6	1.530093	0.000000	0.000000	0.801416	0.000000
chr1	925150	935793	ENST00000437963.5	0.063177	0.000000	0.000000	0.000000	0.000000
chr1	925738	944575	ENST00000342066.7	0.000000	0.000000	0.021964	0.000000	0.133415
chr1	925741	944581	ENST00000617307.4	0.030687	0.000000	0.000000	0.000000	0.000000
chr1	925741	944581	ENST00000618181.4	0.000000	0.000000	0.000000	0.000000	0.000000
chr1	925741	944581	ENST00000622503.4	0.020048	0.021790	1.098797	0.030740	0.101302

GTEx_phenotype_TPM_table.bed

Part 4: eQTLs analysis

4.1 Preparing genotype and phenotype files
(Script: “Prepare_eQTL_input_files.py”)

A. Genotype file

GTEx full genotype table (.vcf, 652 indiv.)

MAF 0.05
remove indels
keep common with breast samples
add ‘chr’ tag
compress & index

GTEx processed genotype table ⋯⋯► genotype Principal Components

script R : “genotype_PCs.R”

B. Phenotype file

224 outputs of GTEx samples

build a table with samples as columns and transcripts expression level as rows.
Script : “make_table_phenotype.py”

GTEx_phenotype_TPM_table.bed

quantile normalization of expression level among samples.
Script : “normalize_phenotype_table.R”

Pheno_table_normalized.bed

order columns according to the genotype file and remove sample not contained in the genotype file.
Script : “order_columns.py”

Pheno_table_normalized.bed

Compute phenotype PEER factors.
Script : “phenotype_PEER_factors.R”

phenotype_PEER_factors.R ⋯⋯ *Script: “ combine_covariates.py”* ⋯⋯►

Gtex.combined_covariates.txt

4.2 eQTL

4.2.1 Permutation pass : get a nominal pvalue for each phenotype

4.2.1.1 Method

Phenotype table

	Sample1	Sample2	Sample3
ENST001	0.1	0.15	0.12
ENST002	0.15	0.2	0.22
ENST003	0.17	0.23	0.15

Permute phenotype table

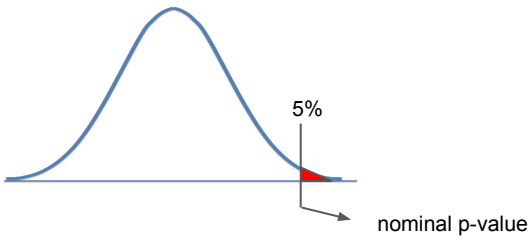
Permuted phenotype table

	Sample1	Sample2	Sample3
ENST001	0.15	0.12	0.1
ENST002	0.2	0.22	0.15
ENST003	0.23	0.15	0.17

\+ *Gtex.combined_covariates.txt*
\+ *genotype table*

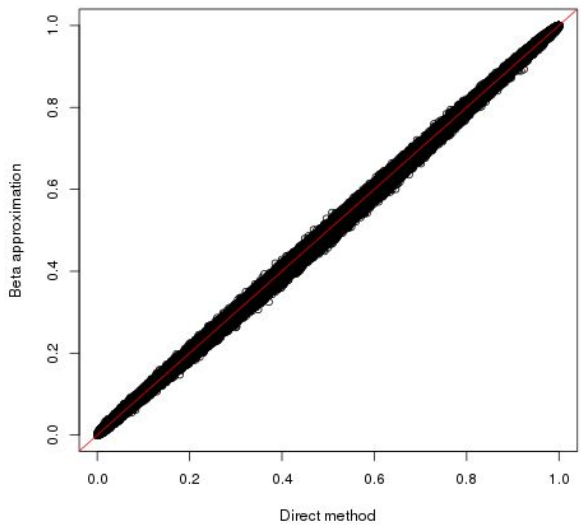
→ *eQTL detection 1000x*

Distribution of empirical p-values for each phenotype.
Compute the nominal p-values for each phenotype.

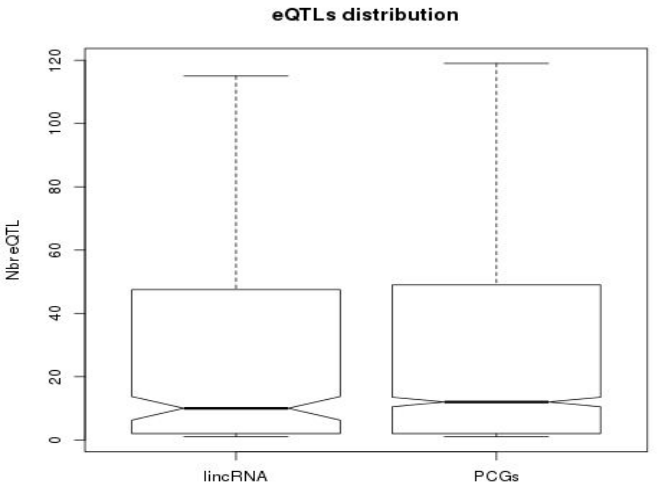


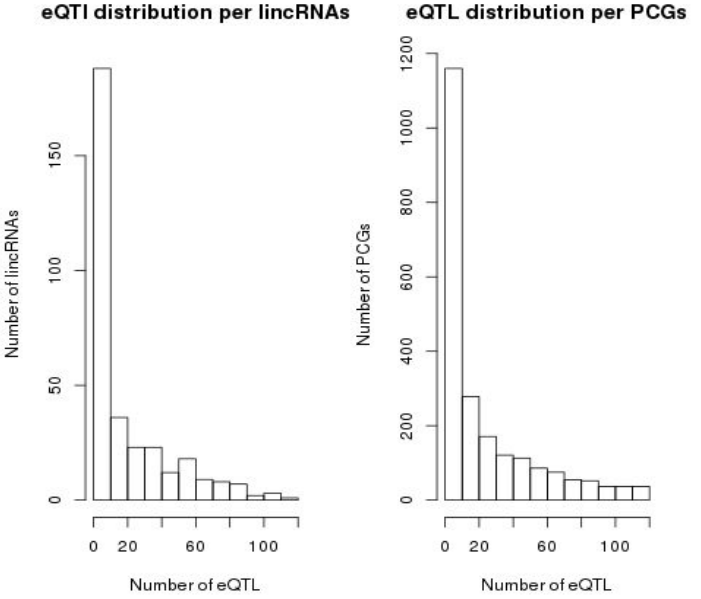
4.2.1.2 Check the beta approximated permutation p-values

We expect to get all points along the diagonal, that shows that unsignificant p-values are well calibrated given the empirical p-values



4.2.2 Conditional pass : get all eQTLs above the nominal p-value





4.3 Get genotype (SNP) affecting both a lincRNA and a cancer transcript

SNP_id	lincRNA(s)_id	PCG(s)_id
17_41464782_A_C_b37	ENST00000632077.1	ENST00000644379.1,ENST00000487825.5
17_41464799_T_G_b37	ENST00000632077.1	ENST00000644379.1,ENST00000487825.5
17_41464827_T_C_b37	ENST00000632077.1	ENST00000644379.1,ENST00000487825.5
6_29937924_T_C_b37	ENST00000630472.1	ENST00000479320.5
17_41464837_A_T_b37	ENST00000632077.1	ENST00000644379.1,ENST00000487825.5
6_29923290_G_T_b37	ENST00000630472.1	ENST00000479320.5
6_29937896_C_T_b37	ENST00000630472.1	ENST00000479320.5
6_29942293_G_C_b37	ENST00000630472.1	ENST00000479320.5
17_41464807_A_G_b37	ENST00000632077.1	ENST00000644379.1,ENST00000487825.5
17_41464774_A_C_b37	ENST00000632077.1	ENST00000644379.1,ENST00000487825.5
6_29921100_C_G_b37	ENST00000630472.1	ENST00000479320.5
17_41382556_G_A_b37	ENST00000632077.1	ENST00000644379.1

coding transcripts:

- ENST00000644379.1 : ENSG00000012048.21
- ENST00000487825.5 : ENSG00000012048.21

BRCA1 (DNA repair associated)

- ENST00000479320.5 : ENSG00000206503.12

HLA-A (major histocompatibility complex, class I, A)

lincRNAs:

- ENST00000632077.1 : ENSG00000267496.4

FAM215A-202 (family with sequence similarity 215 member A)

- ENST00000630472.1 : ENSG00000281831.1

AL645929.2-201