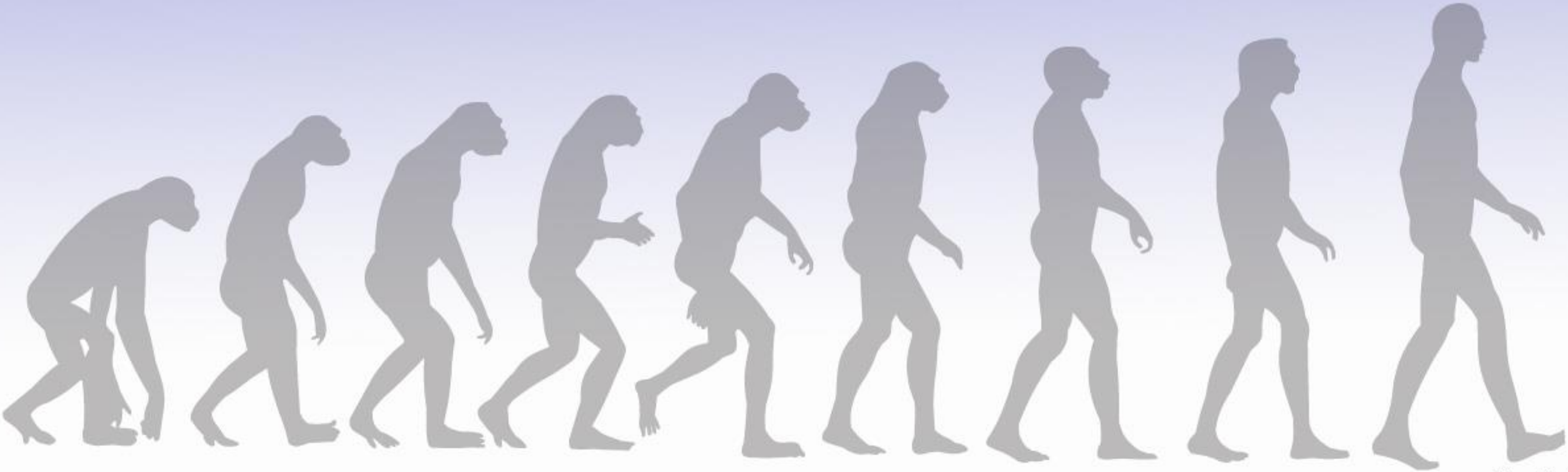
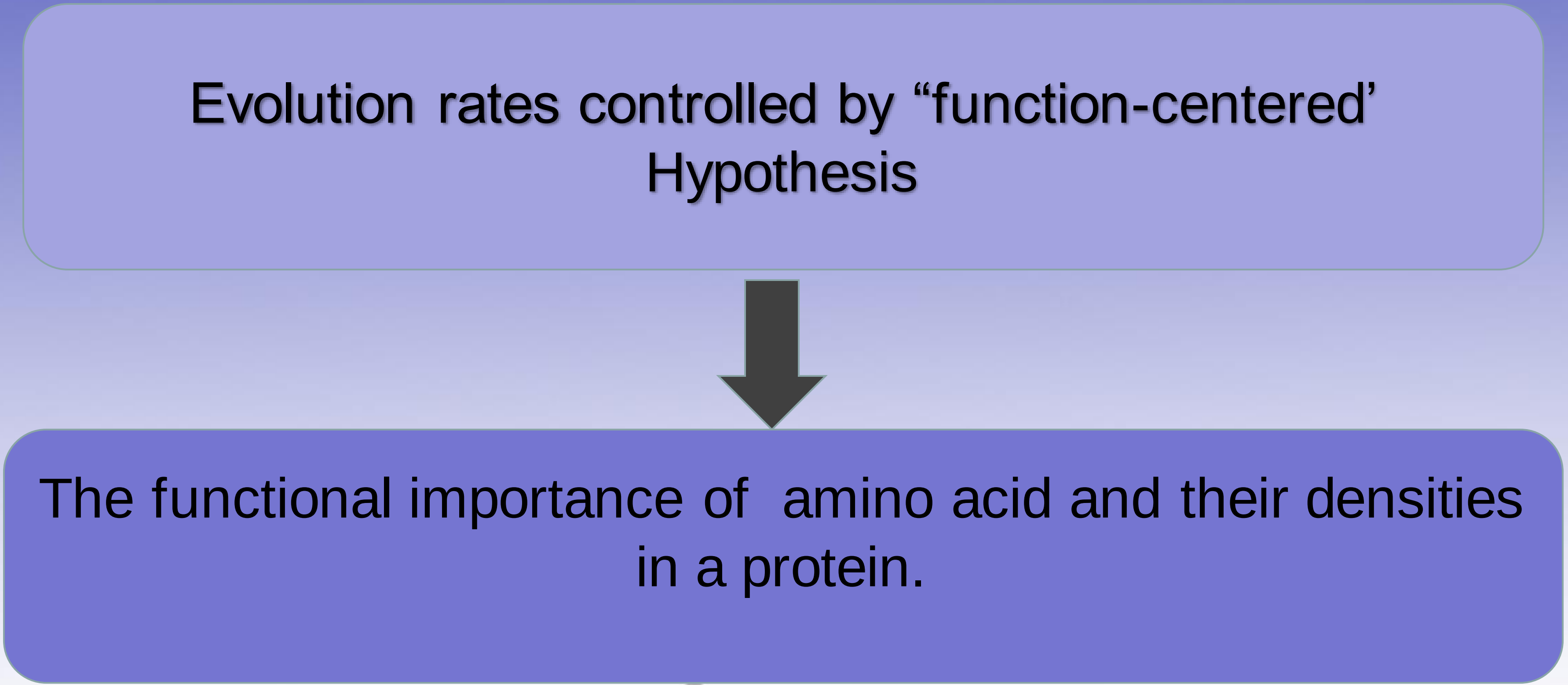
Nile University
جامعة النيل

Gene Expression Level and Single Neuceutide Polymorphism Rates

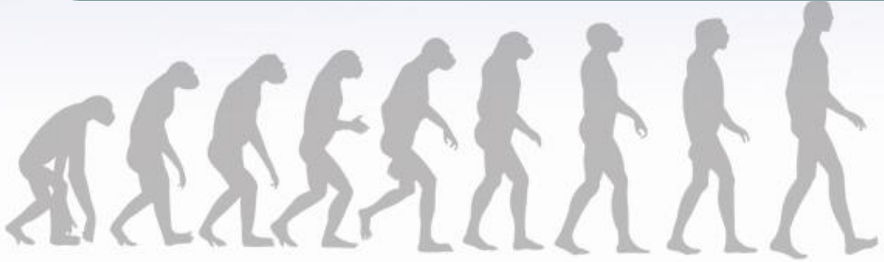


Protein Evolution in the past



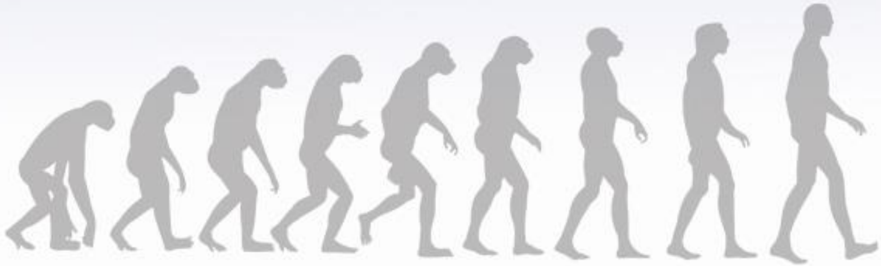
Evolution rates controlled by “function-centered’ Hypothesis

The functional importance of amino acid and their densities in a protein.



Factors affect the proteins evolution

- The genomic position of the encoding genes
- Gene expression patterns
- Position in biological networks and possibly their robustness to mistranslation.



Expression-based evolutionary analysis

Multicellular organisms

Using

Expression breadth

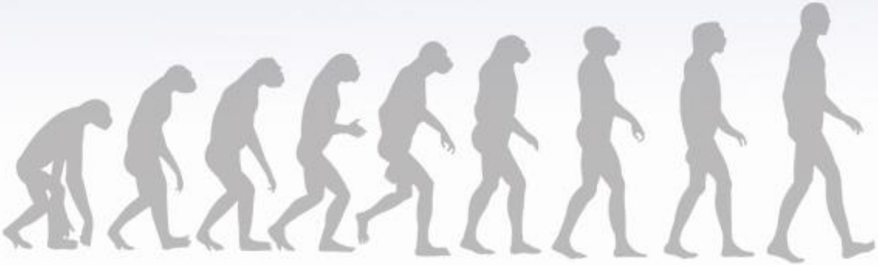
Unicellular organisms

Using

Gene expression

Why “**Expression breadth**” used mainly in Multicellular organisms

- As Genes express at different levels in different tissue types in multicellular organisms.
- Therefore the **Expression breadth** is the number of different tissues where a gene is significantly expressed.

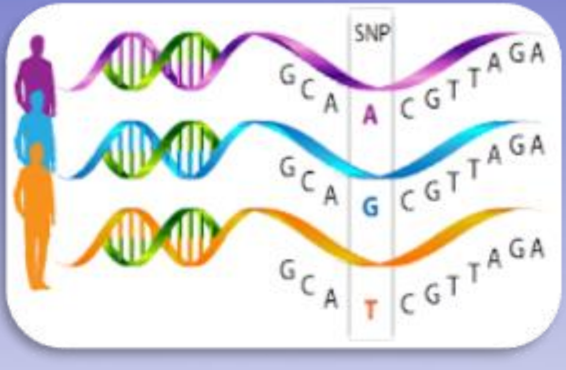


High Vs. Low

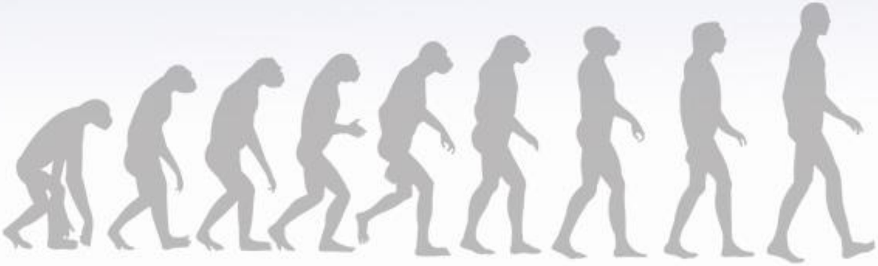
High expressed genes	Low expressed genes
Involved in house-keeping processes	Weakly expressed
Higher and broader expressed, to have more interaction partners	Fewer interaction partners
Evolve slower, and are less prone to gene loss across various taxonomic groups	Genes evolve faster and are more often lost during evolution
Any mutation could be lethal	

Single-nucleotide polymorphisms (SNPs)

- DNA base variants present in the human population at a frequency >1%.

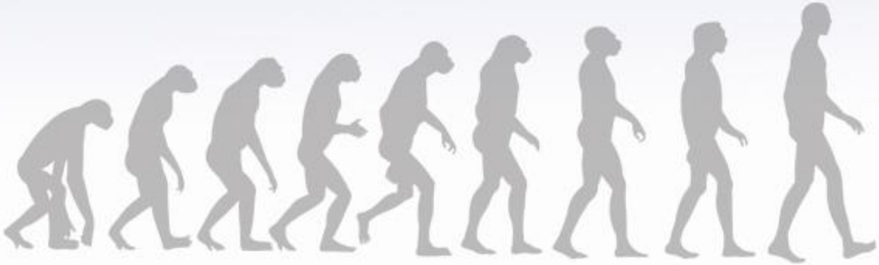


- The non-synonymous coding SNPs & SNPs in regulatory regions have an effect on phenotype.



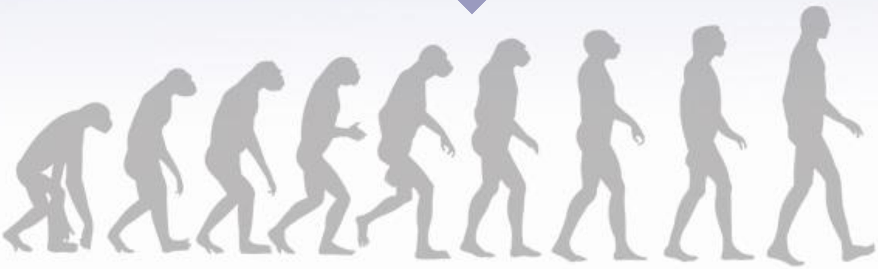
Hypothesis

- Genes that are expressed at higher levels and in a greater number of tissues have lower single nucleotide polymorphism (SNP) rates than genes that are lowly/narrowly expressed.

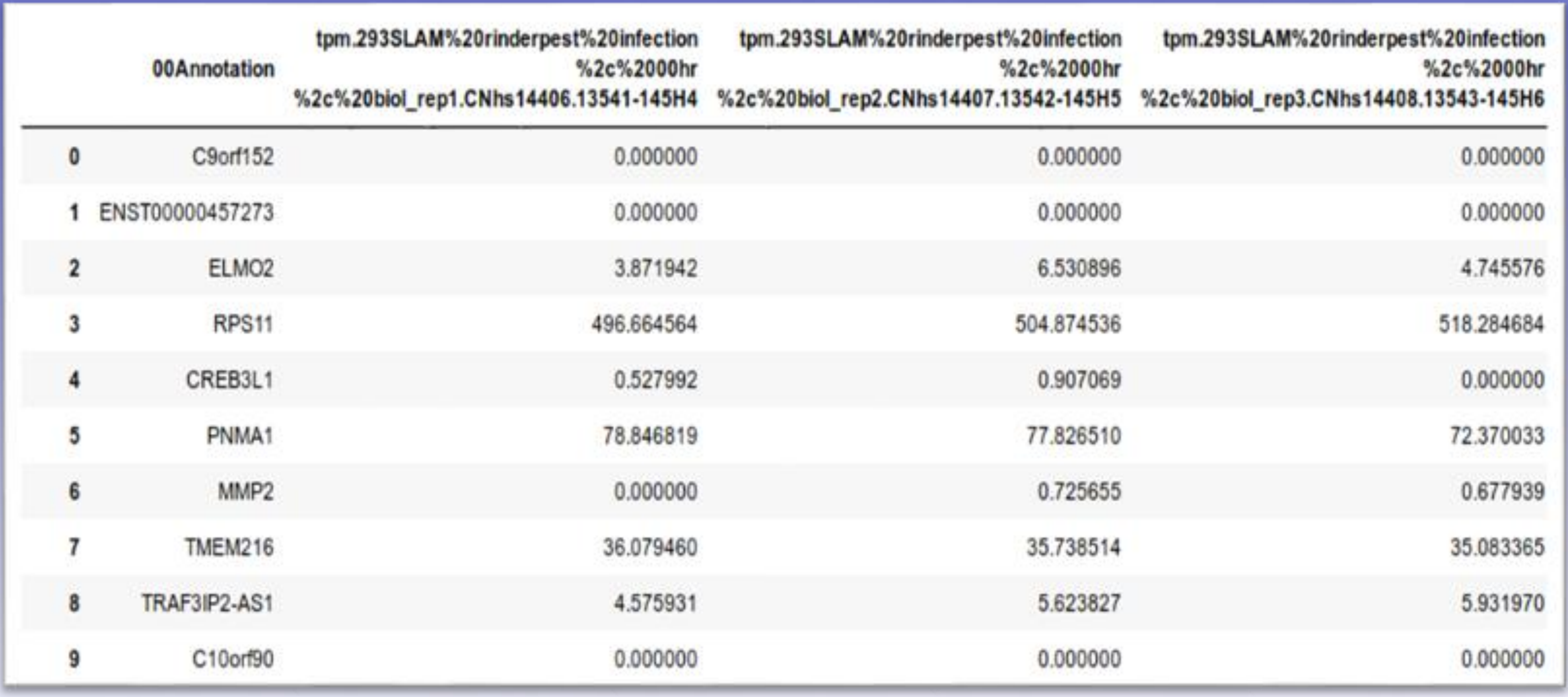


Tools and Methods

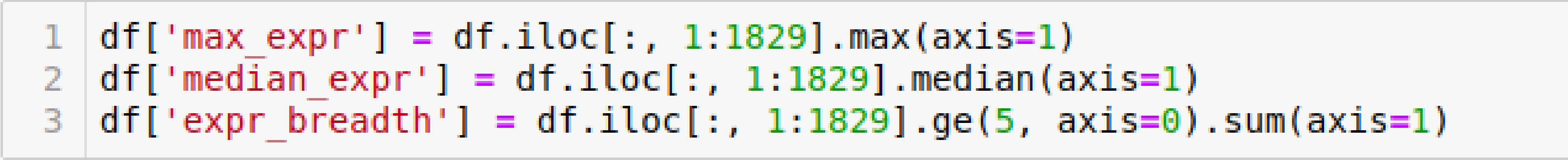
- Mygene
- pybedtools
- Pandas
- Linux command line



Raw Data

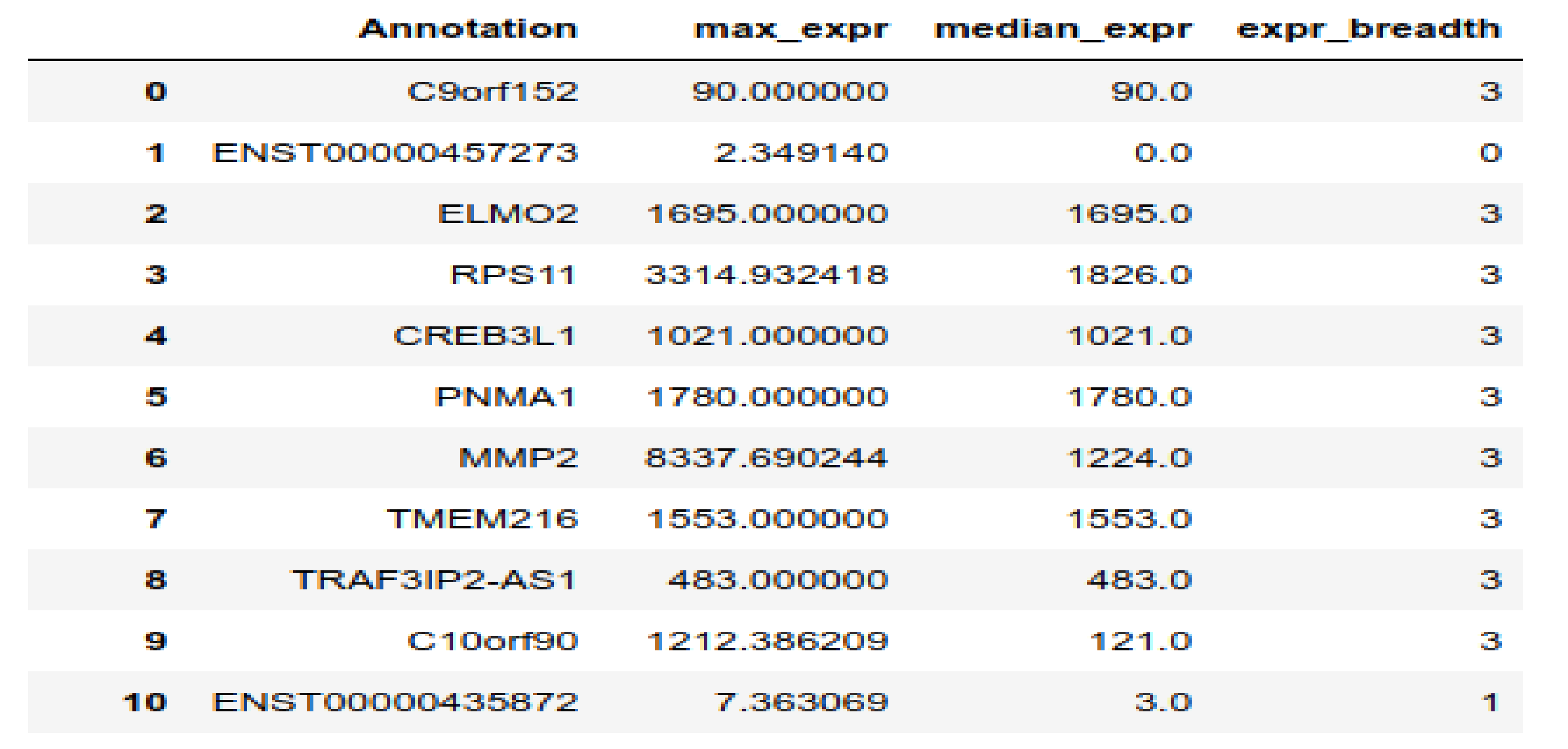


	00Annotation	tpm.293SLAM%20rinderpest%20infection%2c%2000hr%2c%20biol_rep1.CNhs14406.13541-145H4	tpm.293SLAM%20rinderpest%20infection%2c%2000hr%2c%20biol_rep2.CNhs14407.13542-145H5	tpm.293SLAM%20rinderpest%20infection%2c%2000hr%2c%20biol_rep3.CNhs14408.13543-145H6
0	C9orf152	0.000000	0.000000	0.000000
1	ENST00000457273	0.000000	0.000000	0.000000
2	ELMO2	3.871942	6.530896	4.745576
3	RPS11	496.664564	504.874536	518.284684
4	CREB3L1	0.527992	0.907069	0.000000
5	PNMA1	78.846819	77.826510	72.370033
6	MMP2	0.000000	0.725655	0.677939
7	TMEM216	36.079460	35.738514	35.083365
8	TRAF3IP2-AS1	4.575931	5.623827	5.931970
9	C10orf90	0.000000	0.000000	0.000000



```
df['max_expr'] = df.iloc[:, 1:1829].max(axis=1)
df['median_expr'] = df.iloc[:, 1:1829].median(axis=1)
df['expr_breadth'] = df.iloc[:, 1:1829].ge(5, axis=0).sum(axis=1)
```

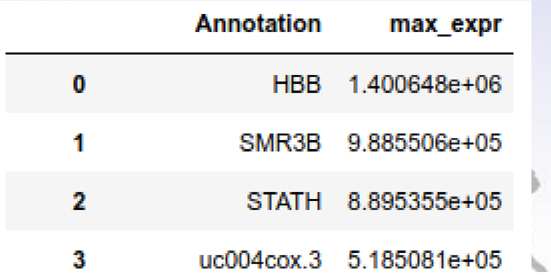
Max-Median-Breadth Calculations



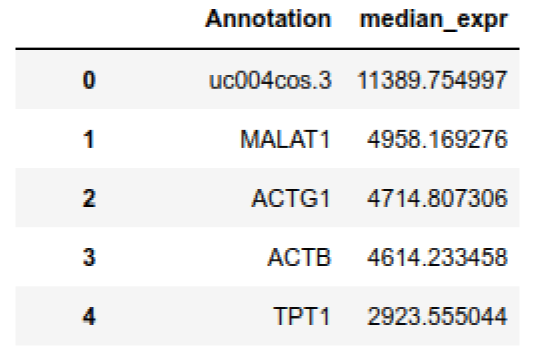
	Annotation	max_expr	median_expr	expr_breadth
0	C9orf152	90.000000	90.0	3
1	ENST00000457273	2.349140	0.0	0
2	ELMO2	1695.000000	1695.0	3
3	RPS11	3314.932418	1826.0	3
4	CREB3L1	1021.000000	1021.0	3
5	PNMA1	1780.000000	1780.0	3
6	MMP2	8337.690244	1224.0	3
7	TMEM216	1553.000000	1553.0	3
8	TRAF3IP2-AS1	483.000000	483.0	3
9	C10orf90	1212.386209	121.0	3
10	ENST00000435872	7.363069	3.0	1

High/Low genes selection

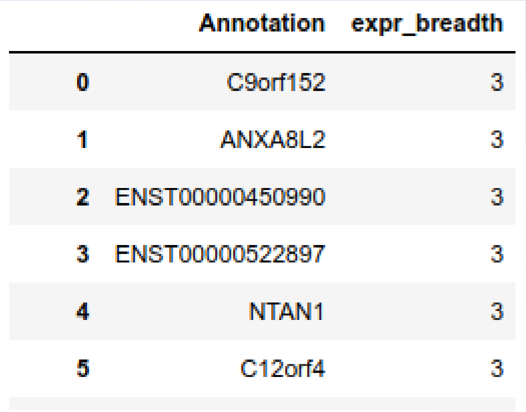
- Sorting genes according to these values:
  - Max Expression.
  - Median Expression.
  - Expression Breadth.
- Selecting top 5% and low 5% of genes according to the calculated values.



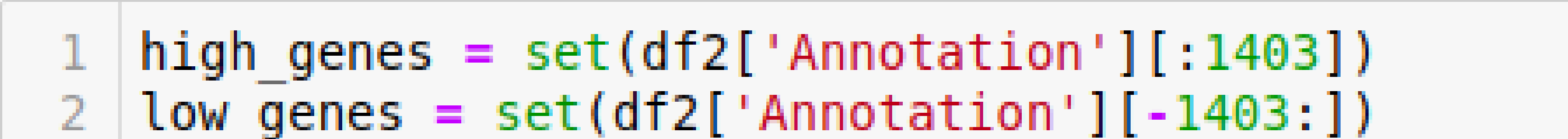
	Annotation	max_expr
0	HBB	1.400648e+06
1	SMR3B	9.885506e+05
2	STATH	8.895355e+05
3	uc004cox.3	5.185081e+05



	Annotation	median_expr
0	uc004cos.3	11389.754997
1	MALAT1	4958.169276
2	ACTG1	4714.807306
3	ACTB	4614.233458
4	TPT1	2923.555044

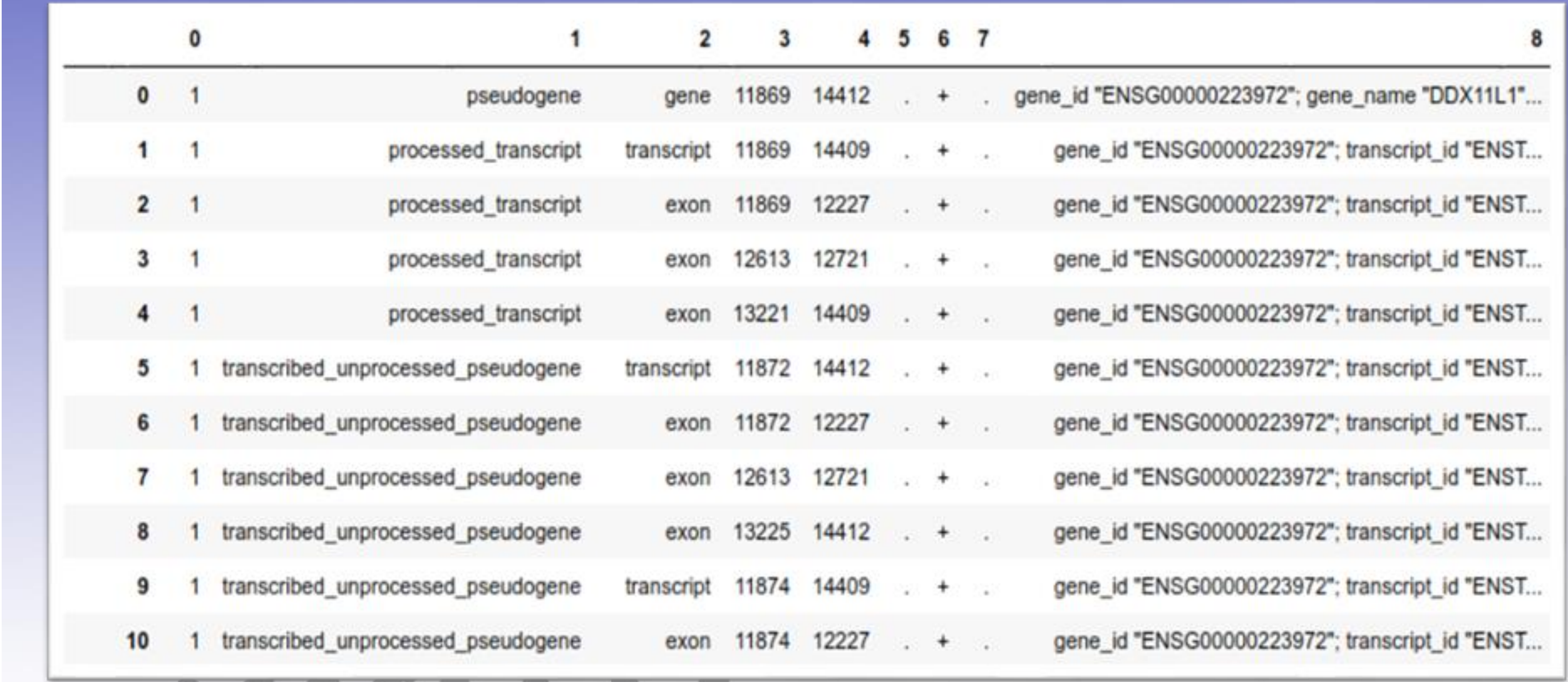


	Annotation	expr_breadth
0	C9orf152	3
1	ANXA8L2	3
2	ENST00000450990	3
3	ENST00000522897	3
4	NTAN1	3
5	C12orf4	3
		3
		3
		3



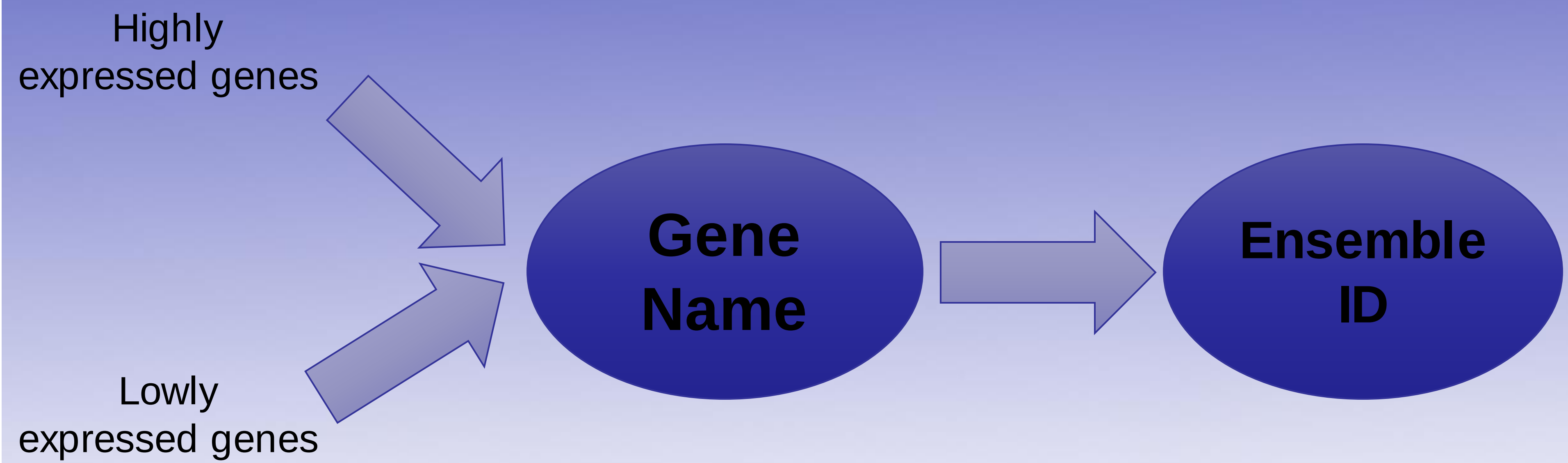
```
high_genes = set(df2['Annotation'][:1403])
low_genes = set(df2['Annotation'][-1403:])
```

Gene Transfer Format of the hg19/GRCh37 genome



	0	1	2	3	4	5	6	7	8
0	1	pseudogene	gene	11869	14412	.	+	.	gene_id "ENSG00000223972"; gene_name "DDX11L1"...
1	1	processed_transcript	transcript	11869	14409	.	+	.	gene_id "ENSG00000223972"; transcript_id "ENST...
2	1	processed_transcript	exon	11869	12227	.	+	.	gene_id "ENSG00000223972"; transcript_id "ENST...
3	1	processed_transcript	exon	12613	12721	.	+	.	gene_id "ENSG00000223972"; transcript_id "ENST...
4	1	processed_transcript	exon	13221	14409	.	+	.	gene_id "ENSG00000223972"; transcript_id "ENST...
5	1	transcribed_unprocessed_pseudogene	transcript	11872	14412	.	+	.	gene_id "ENSG00000223972"; transcript_id "ENST...
6	1	transcribed_unprocessed_pseudogene	exon	11872	12227	.	+	.	gene_id "ENSG00000223972"; transcript_id "ENST...
7	1	transcribed_unprocessed_pseudogene	exon	12613	12721	.	+	.	gene_id "ENSG00000223972"; transcript_id "ENST...
8	1	transcribed_unprocessed_pseudogene	exon	13225	14412	.	+	.	gene_id "ENSG00000223972"; transcript_id "ENST...
9	1	transcribed_unprocessed_pseudogene	transcript	11874	14409	.	+	.	gene_id "ENSG00000223972"; transcript_id "ENST...
10	1	transcribed_unprocessed_pseudogene	exon	11874	12227	.	+	.	gene_id "ENSG00000223972"; transcript_id "ENST...

Gene Names to Ensemble Gene IDs Conversion

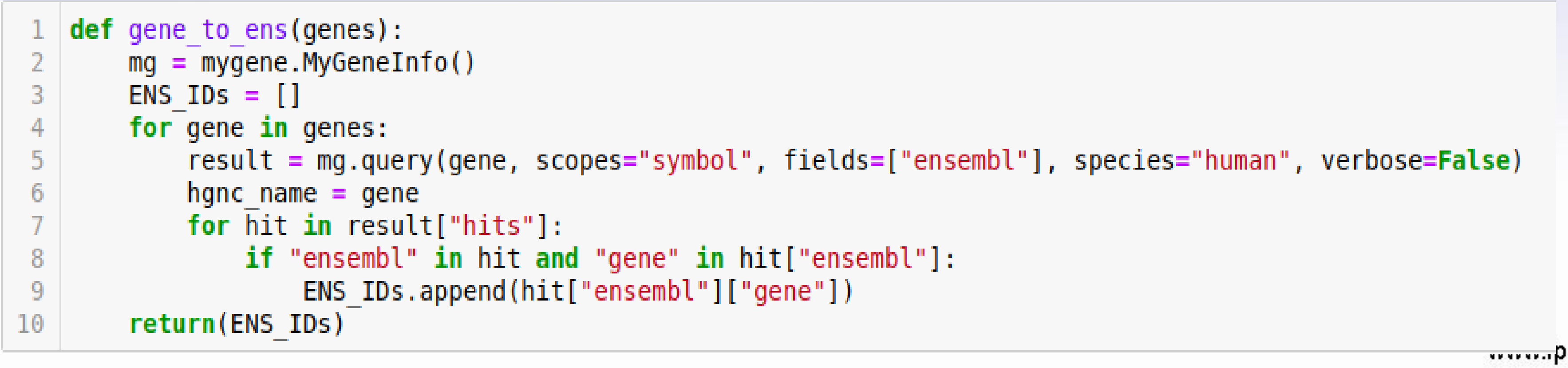


Highly expressed genes

Lowly expressed genes

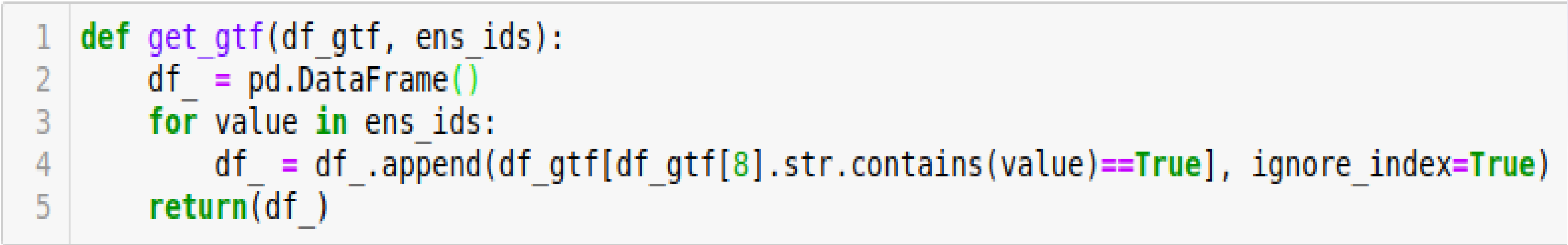
Gene Name

Ensemble ID



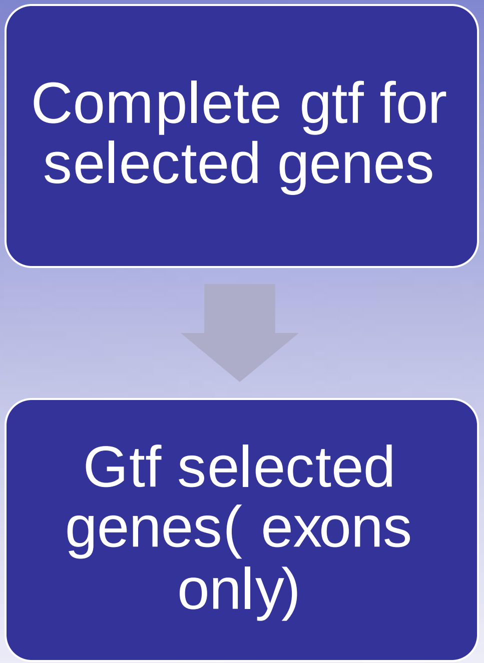
```python
def gene_to_ens(genes):
    mg = mygene.MyGeneInfo()
    ENS_IDs = []
    for gene in genes:
        result = mg.query(gene, scopes="symbol", fields=["ensembl"], species="human", verbose=False)
        hgnc_name = gene
        for hit in result["hits"]:
            if "ensembl" in hit and "gene" in hit["ensembl"]:
                ENS_IDs.append(hit["ensembl"]["gene"])
    return(ENS_IDs)
```

Gtf file for selected genes



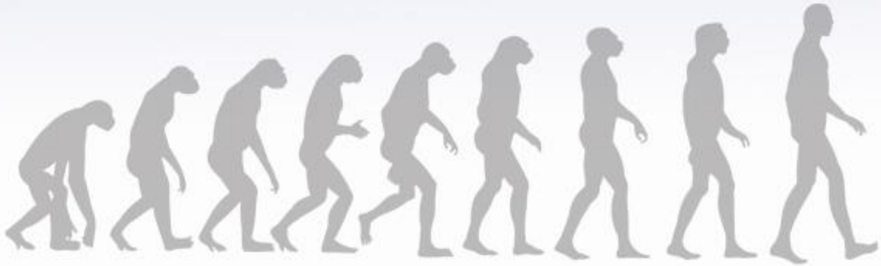
```
def get_gtf(df_gtf, ens_ids):
    df_ = pd.DataFrame()
    for value in ens_ids:
        df_ = df_.append(df_gtf[df_gtf[8].str.contains(value)==True], ignore_index=True)
    return(df_)
```

Where is the coding regions????

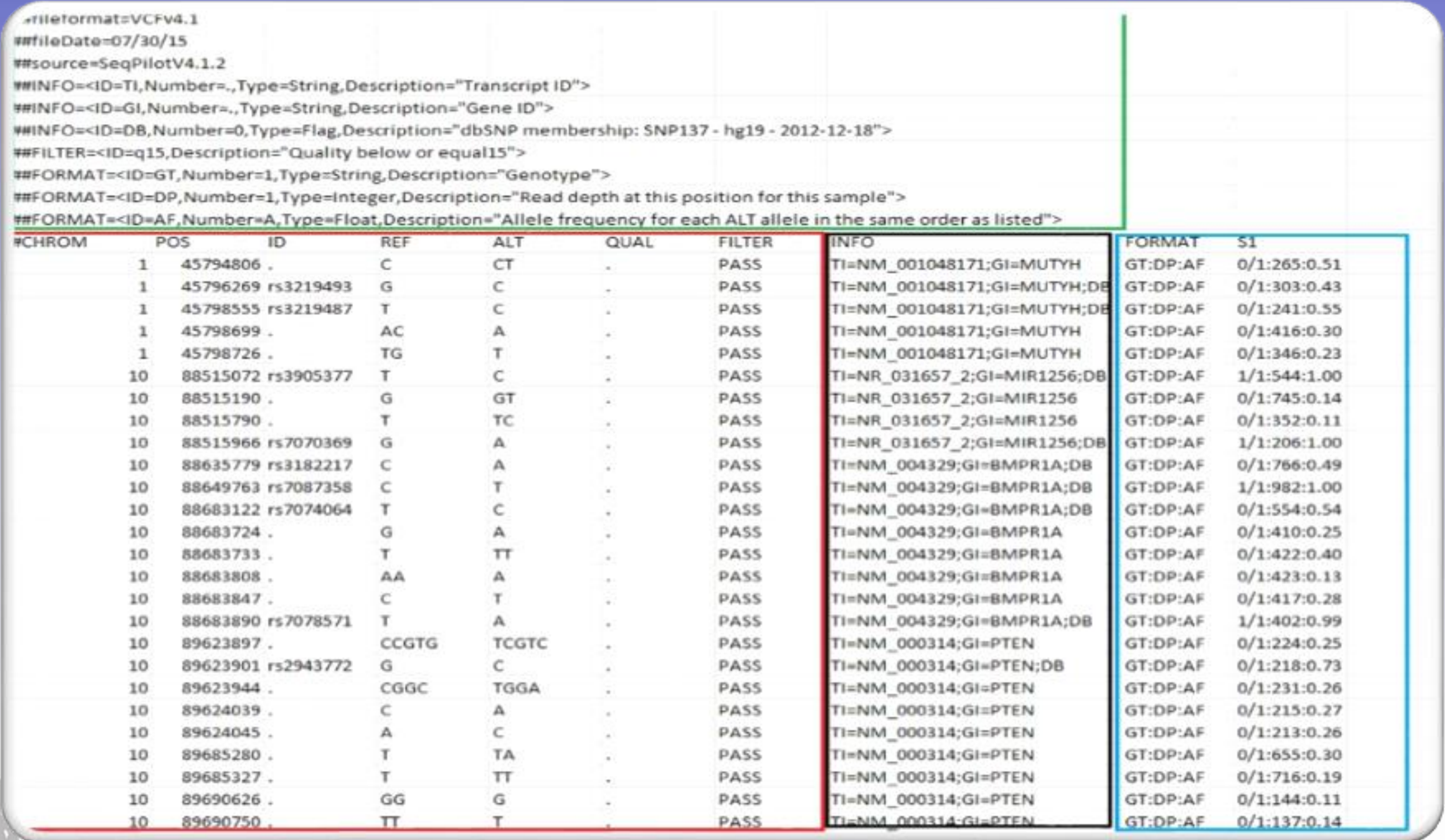


Complete gtf for selected genes

Gtf selected genes(exons only)



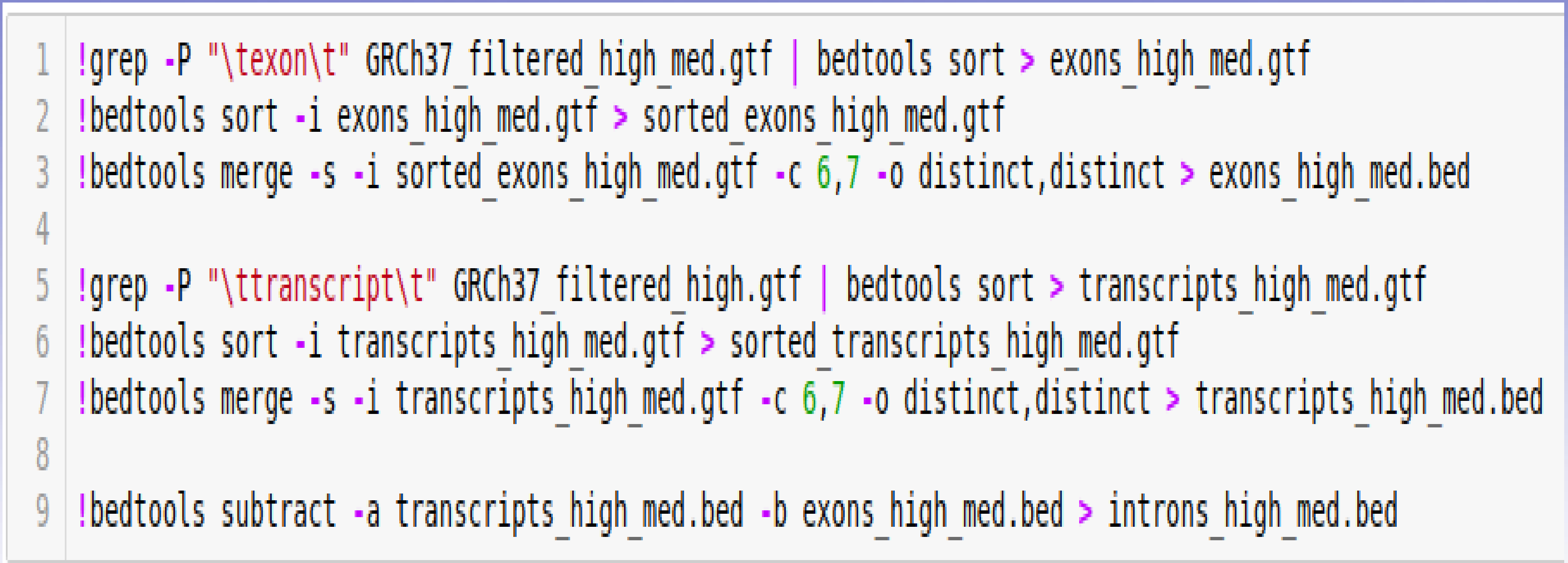
SNPs file (VCF file)



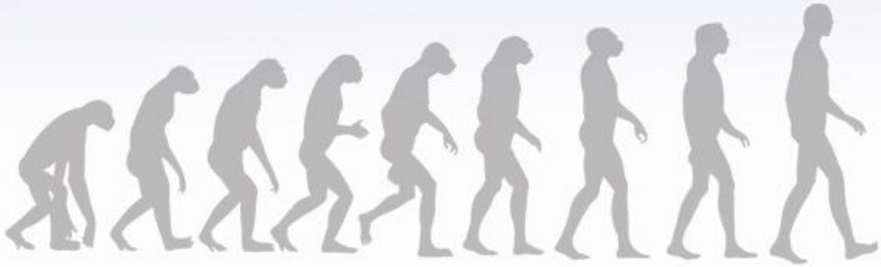
```
##fileformat=VCFv4.1
##fileDate=07/30/15
##source=SeqPilotV4.1.2
##INFO=<ID=TI,Number=.,Type=String,Description="Transcript ID">
##INFO=<ID=GI,Number=.,Type=String,Description="Gene ID">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership: SNP137 - hg19 - 2012-12-18">
##FILTER=<ID=q15,Description="Quality below or equal15">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read depth at this position for this sample">
##FORMAT=<ID=AF,Number=A,Type=Float,Description="Allele frequency for each ALT allele in the same order as listed">
```

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	S1
1	45794806	.	C	CT	.	PASS	TI=NM_001048171;GI=MUTYH	GT:DP:AF	0/1:265:0.51
1	45796269	rs3219493	G	C	.	PASS	TI=NM_001048171;GI=MUTYH;DB	GT:DP:AF	0/1:303:0.43
1	45798555	rs3219487	T	C	.	PASS	TI=NM_001048171;GI=MUTYH;DB	GT:DP:AF	0/1:241:0.55
1	45798699	.	AC	A	.	PASS	TI=NM_001048171;GI=MUTYH	GT:DP:AF	0/1:416:0.30
1	45798726	.	TG	T	.	PASS	TI=NM_001048171;GI=MUTYH	GT:DP:AF	0/1:346:0.23
10	88515072	rs3905377	T	C	.	PASS	TI=NR_031657_2;GI=MIR1256;DB	GT:DP:AF	1/1:544:1.00
10	88515190	.	G	GT	.	PASS	TI=NR_031657_2;GI=MIR1256	GT:DP:AF	0/1:745:0.14
10	88515790	.	T	TC	.	PASS	TI=NR_031657_2;GI=MIR1256	GT:DP:AF	0/1:352:0.11
10	88515966	rs7070369	G	A	.	PASS	TI=NR_031657_2;GI=MIR1256;DB	GT:DP:AF	1/1:206:1.00
10	88635779	rs3182217	C	A	.	PASS	TI=NM_004329;GI=BMPR1A;DB	GT:DP:AF	0/1:766:0.49
10	88649763	rs7087358	C	T	.	PASS	TI=NM_004329;GI=BMPR1A;DB	GT:DP:AF	1/1:982:1.00
10	88683122	rs7074064	T	C	.	PASS	TI=NM_004329;GI=BMPR1A;DB	GT:DP:AF	0/1:554:0.54
10	88683724	.	G	A	.	PASS	TI=NM_004329;GI=BMPR1A	GT:DP:AF	0/1:410:0.25
10	88683733	.	T	TT	.	PASS	TI=NM_004329;GI=BMPR1A	GT:DP:AF	0/1:422:0.40
10	88683808	.	AA	A	.	PASS	TI=NM_004329;GI=BMPR1A	GT:DP:AF	0/1:423:0.13
10	88683847	.	C	T	.	PASS	TI=NM_004329;GI=BMPR1A	GT:DP:AF	0/1:417:0.28
10	88683890	rs7078571	T	A	.	PASS	TI=NM_004329;GI=BMPR1A;DB	GT:DP:AF	1/1:402:0.99
10	89623897	.	CCGTG	TCGTC	.	PASS	TI=NM_000314;GI=PTEN	GT:DP:AF	0/1:224:0.25
10	89623901	rs2943772	G	C	.	PASS	TI=NM_000314;GI=PTEN;DB	GT:DP:AF	0/1:218:0.73
10	89623944	.	CGGC	TGGA	.	PASS	TI=NM_000314;GI=PTEN	GT:DP:AF	0/1:231:0.26
10	89624039	.	C	A	.	PASS	TI=NM_000314;GI=PTEN	GT:DP:AF	0/1:215:0.27
10	89624045	.	A	C	.	PASS	TI=NM_000314;GI=PTEN	GT:DP:AF	0/1:213:0.26
10	89685280	.	T	TA	.	PASS	TI=NM_000314;GI=PTEN	GT:DP:AF	0/1:655:0.30
10	89685327	.	T	TT	.	PASS	TI=NM_000314;GI=PTEN	GT:DP:AF	0/1:716:0.19
10	89690626	.	GG	G	.	PASS	TI=NM_000314;GI=PTEN	GT:DP:AF	0/1:144:0.11
10	89690750	.	TT	T	.	PASS	TI=NM_000314;GI=PTEN	GT:DP:AF	0/1:137:0.14

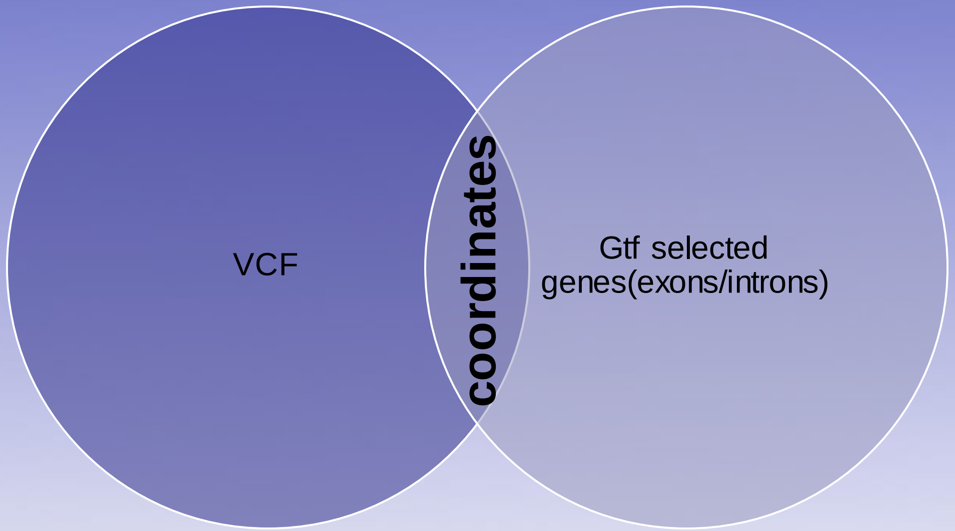
Get Exons/ Introns

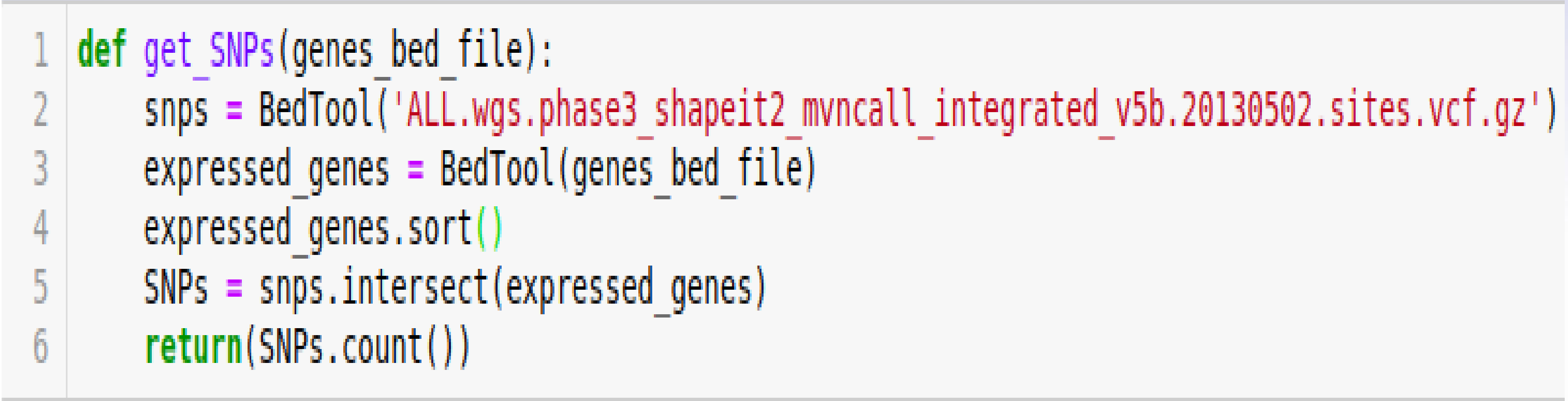


```
!grep -P "\texon\t" GRCh37_filtered_high_med.gtf | bedtools sort > exons_high_med.gtf
!bedtools sort -i exons_high_med.gtf > sorted_exons_high_med.gtf
!bedtools merge -s -i sorted_exons_high_med.gtf -c 6,7 -o distinct,distinct > exons_high_med.bed

!grep -P "\ttranscript\t" GRCh37_filtered_high.gtf | bedtools sort > transcripts_high_med.gtf
!bedtools sort -i transcripts_high_med.gtf > sorted_transcripts_high_med.gtf
!bedtools merge -s -i transcripts_high_med.gtf -c 6,7 -o distinct,distinct > transcripts_high_med.bed

!bedtools subtract -a transcripts_high_med.bed -b exons_high_med.bed > introns_high_med.bed
```



Intersection

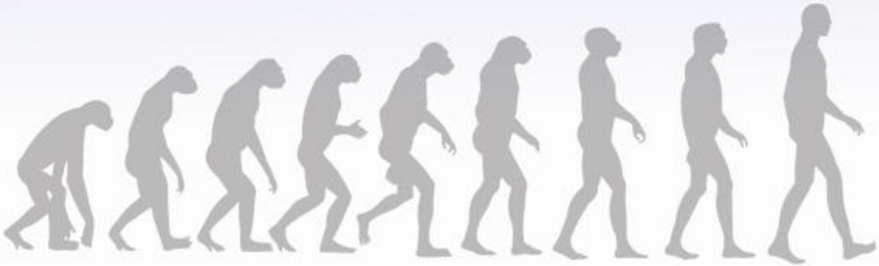




```python
def get_SNPs(genes_bed_file):
    snps = BedTool('ALL.wgs.phase3_shapeit2_mvncall_integrated_v5b.20130502.sites.vcf.gz')
    expressed_genes = BedTool(genes_bed_file)
    expressed_genes.sort()
    SNPs = snps.intersect(expressed_genes)
    return(SNPs.count())
```

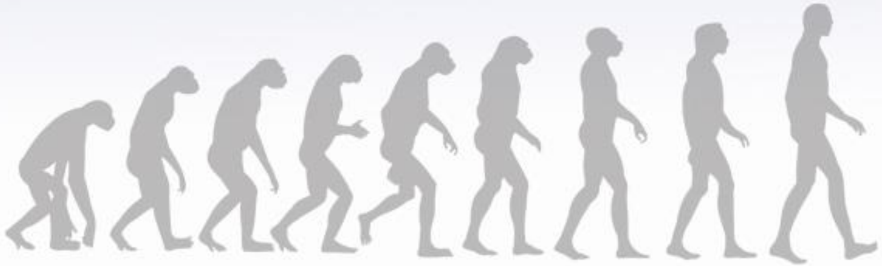
Results (exons)

	Max Expression	Median Expression	Breadth Expression
Highly Expressed Genes	0.0034899837337915773	0.0031972270987368073	0.003228856553583139
Lowly Expressed genes	0.0005007792375417857	0.001423264075509528	0.028695393161306672



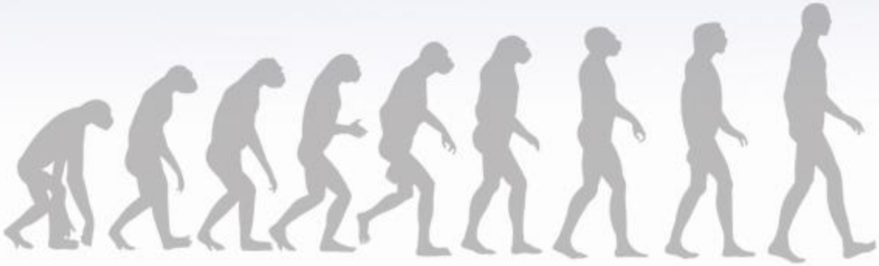
Results (introns)

	Max Expression	Median Expression	Breadth Expression
Highly Expressed Genes	0.03314763184341043	0.02827210363243538	0.0005027683569847616
Lowly Expressed genes	0.009858628492566827	0.023731668376511285	0.009577278598021467



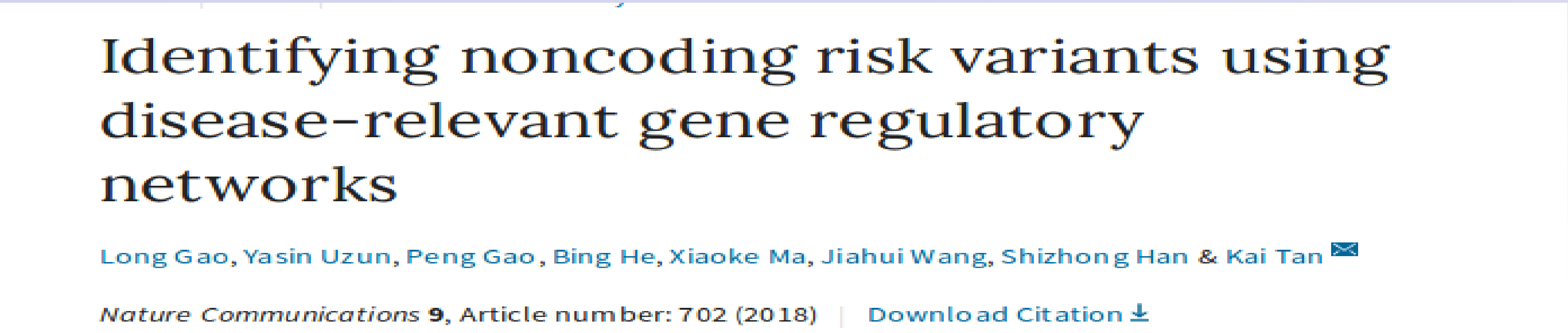
Conclusion

- According to exons: SNPs frequency inversely proportional gene expression mainly according to breadth
- While according to introns: SNPs frequency inversely proportional gene expression mainly according to breadth



What else ….?

- We can conclude a **new hypothesis** that the SNPs in noncoding regions can associated with number of diseases, this improves our understanding of noncoding of genomes and their roles in disease.



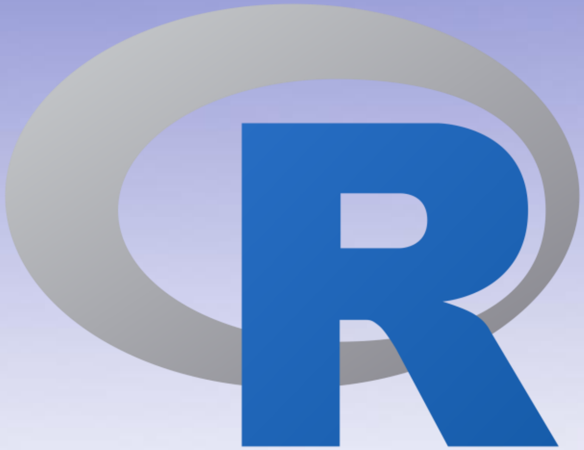
Identifying noncoding risk variants using disease-relevant gene regulatory networks

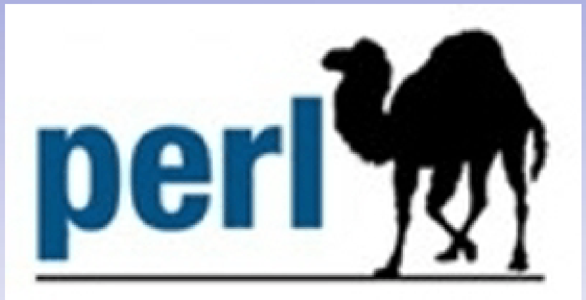
Long Gao, Yasin Uzun, Peng Gao, Bing He, Xiaoke Ma, Jiahui Wang, Shizhong Han & Kai Tan

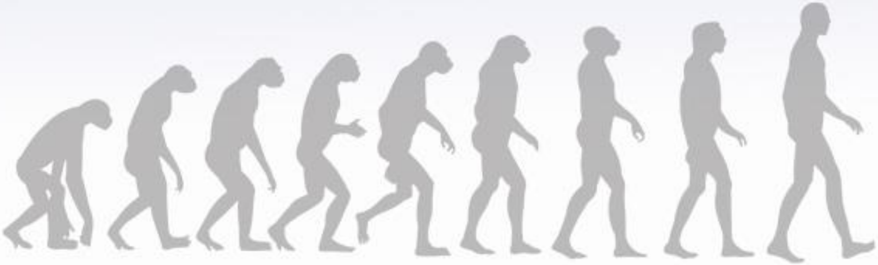
Nature Communications **9**, Article number: 702 (2018) | Download Citation

Recommendation

- This all steps can be done using :







THANK YOU