CS3047701 資料科學導論期末報告

癌症病人生存預測

張秋霞　鄭妮妮

一・實作介紹

近年來，癌症的確診率和死亡率一直在增加。國際抗癌聯盟（UICC）曾指出，全世界每年因癌症導致的死亡人數超過 800 萬，這幾乎相當於一整個紐約市的人口。隨著癌症病人的增加，對癌症病人的生存情況進行精準分析也變得尤爲重要。爲了研究癌症病人的生存情況，僅僅根據臨床的資訊是遠遠不夠的，若加入基因資訊來對病人進行生存分析，人體龐大基因庫將使得整個數據集的維度變得特別高，因此，我們嘗試在本次期末實作中探索一種能夠有效提取出癌症相關基因，並結合基因資訊和臨床資訊對癌症病人進行生存分析的預測方法。在本次實作中，我們選用癌症病人資訊較多的公開乳癌資料集 TCGA-BRCA，利用 edgeR 和 DESeq 篩選出乳癌相關的差異基因，並結合乳癌病人臨床資訊，利用生存預測模型 Deepsurv 對乳癌病人進行生存預測。
實作程式碼可點擊鏈接查看：https://github.com/ChoushaChang/P-Deepsurv

二・相關研究

2.1 差異基因

差異基因（differential gene），是指在不同因素下由於基因突變或者甲基化等結構發生改變導致與正常組織基因有差異的基因。差異基因的表達和正常基因會有有顯著性差異。

2.2 edgeR

edgeR[1,2]是一個可用於差異基因分析的 R 語言分析套件，它基於負二項式分佈將新的統計方法實現爲計數變異性模型，包括經驗貝葉斯方法，精確檢驗和廣義線性模型。該套件特別適合分析具有多個實驗因素但可能有少量重複的設計實驗，即使在小樣本中，它也具有對轉錄本特異性變異進行建模的獨特能力，這對於區分在復製過程中具有一致作用的基因或轉錄本的優先級至關重要，該套件的具體實作原理可參考[2]。

2.3 DESeq

DESeq[3]是一個 R 語言分析套件，常用於分析 RNA-seq 的資料和基因表現量差異。模擬基因在隨機抽樣定序中，按照負二項分佈的假設對數據進行估計的分佈，並且依據 FDR(false discovery rate)計算各個基因間是否有顯著的差異。DESeq 的特性在於它可以估計局部的變異，對於不同基因表現量採用不同的變異參數，可以降低因爲高表現量所產生的偏見，得到更加準確的結果，此套件的具體原理課參考[3]。

2.4 生存分析 & Deepsurv

生存分析（Survival analysis）是指根據試驗或調查得到的數據對生物或人的生存時

間進行分析和推斷，研究生存時間和結局與眾多影響因素間關係及其程度大小的方法，也稱生存率分析或存活率分析。

傳統的用於生存分析的方法是採用線性的 Cox 比例風險（linear Cox proportional hazards model, CPH）模型[4]，但是這需要通過廣泛的特徵工程或者現有的醫學知識來進行特徵篩選，而採用非線性的生存方法，例如神經網路，可以不用通過人工的特徵工程，能夠自動地對這些特徵進行建模，但是傳統的神經網絡模型[5,6]，如時間編碼[7,8]、風險預測[9]方法等，效果往往沒有普通的 CPH 模型好。Deepsurv [10]是 2018 年提出的 Cox 比例風險深度神經網路，它結合了深度神經網路和 CPH 的優點，取得了比 CPH 更好的效果，因此我們在此次實作中選用 Deepsurv 進行生存分析。

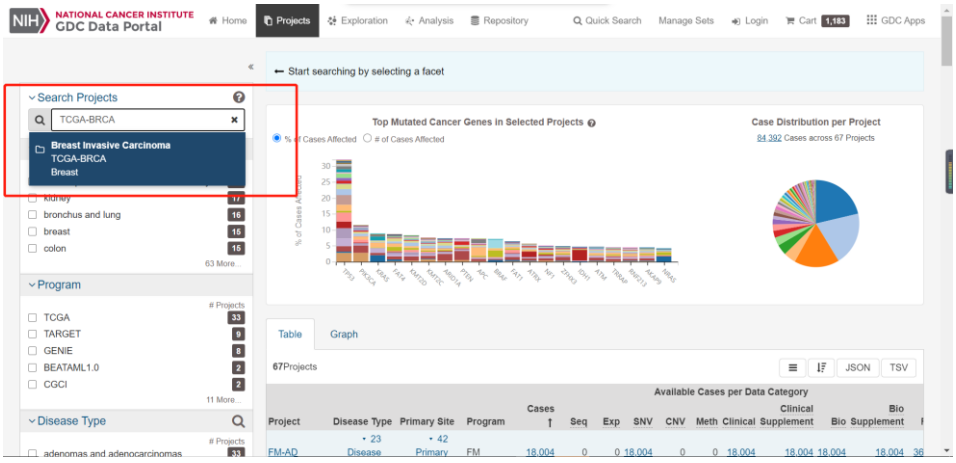
2.5 C-index

C-index（concordance index）用於估計模型的預測能力。C-index 的具體方式爲把資料及研究的所有物件隨機兩兩配對，計算所有對子中預測結果與實際結果一致的對子所占的比例。以生存分析爲例，隨機選出兩位病人，如果真實生存時間較長的一位的預測生存時間也比另一位預測生存時間更長，則代表預測結果與實際結果一致。因此 C-index 結果會在 0.5-1 之間，實際應用中 0.5-0.7 被認爲是較低的準確度，而 0.9 以上則被認爲具有高準確度。在臨床的應用中對於模型真實值與預測值之間的差異分析更加有效，使用的也更加廣泛，C-index 就是此種分析指數。

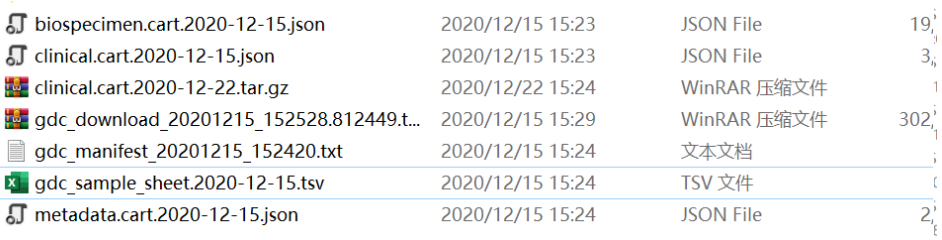
三・實作過程

3.1 資料集下載與處理

首先從資料集官方網站 https://portal.gdc.cancer.gov/projects 下載最新資料集。

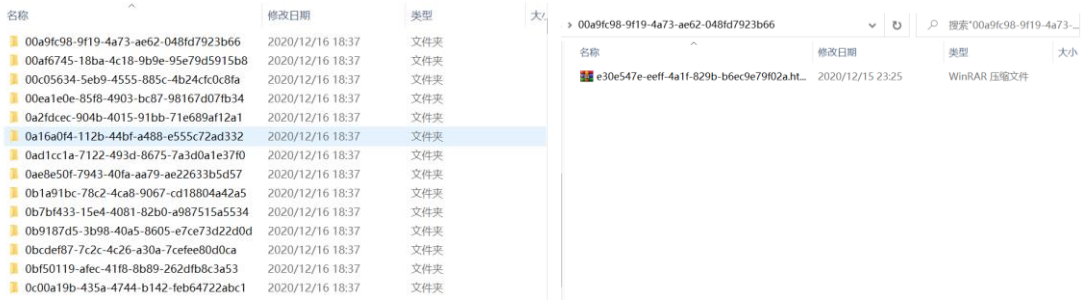


資料下載示意圖



下載所得資料

最終將得到一個包含各個病人資訊的 metadata json 檔以及 1222 個按照病人癌症樣本資訊分類的資料夾，每個資料夾中包含一個壓縮包，壓縮包中含有包含病人基因表達資訊的 counts 檔。

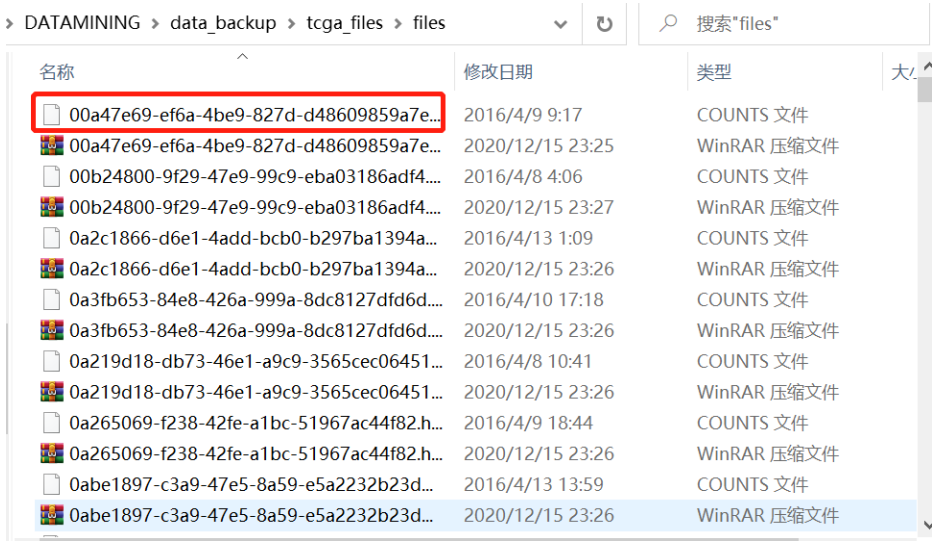


資料夾及資料夾下的壓縮檔

使用 ‘perl putFilesToOneDir.pl’ 執行 putFilesToOneDir.pl，將所有資料夾中的壓縮檔整合到同一個資料夾中，並將其解壓。

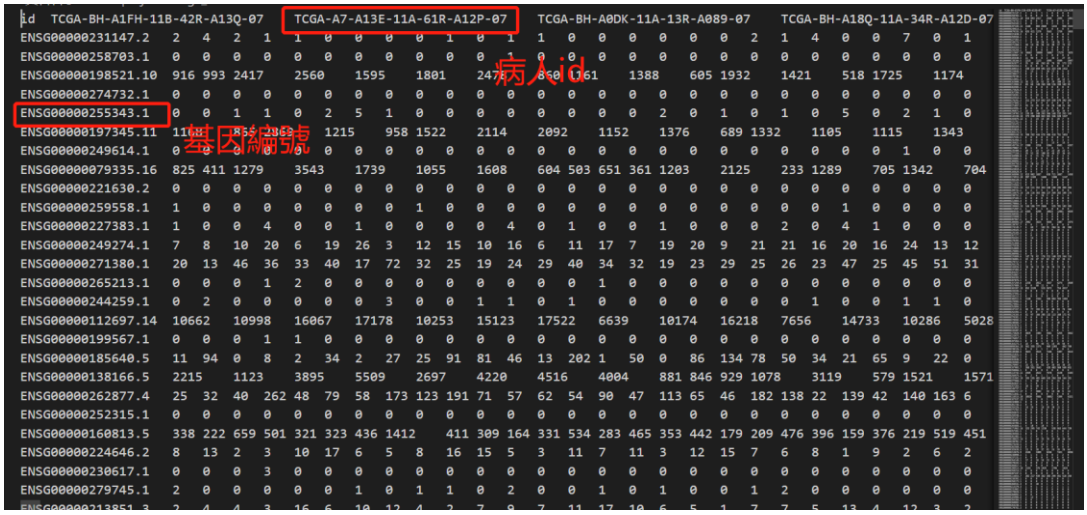
```perl
use strict;
use warnings;

my $newDir="files";
unless(-d $newDir)
{
    mkdir $newDir or die $!;
}

my @allFiles=glob("*");
foreach my $subDir(@allFiles)
{
    if(-d $subDir)
    {
        opendir(SUB,".\\$subDir") or die $!;
        while(my $file=readdir(SUB))
        {
            if($file=~/\.gz$/)
            {
                `copy .\\$subDir\\$file .\\$newDir`;
            }
        }
        close(SUB);
    }
}
```

putFilesToOneDir.pl 程式碼



解壓後得到 counts 檔

使用‘perl mRNA_merge.pl metadata.cart.2020-12-15.json’ 執行 mRNA_merge.pl，該程式碼結合 metadata.json 將基因表達資訊(mRNA 資訊)提取出來，整理到 mRNAmatrix.txt 檔中，並返回正常組織樣本數及癌變組織樣本數。



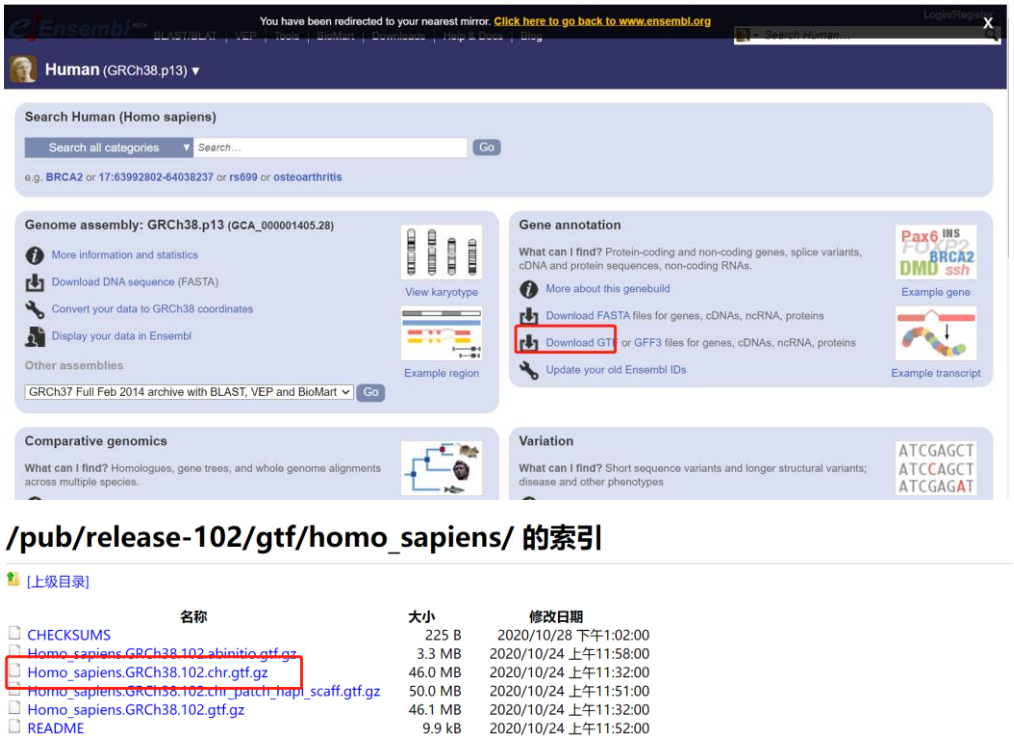
mRNAmatrix.txt 檔內容

```
for my $i(@{$obj})
{
        my $file_name=$i->{'file_name'};
      my $file_id=$i->{'file_id'};
      my $entity_submitter_id=$i->{'associated_entities'}->[0]->{'entity_submitter_id'};
      $file_name=~s/\.gz//g;
      if(-f $file_name)
      {
         my @idArr=split(/\-/,$entity_submitter_id);
         if($idArr[3]=~/^0/)
         {
             push(@tumorSamples,$entity_submitter_id);
         }
         else
         {
           push(@normalSamples,$entity_submitter_id);
        }
         open(RF,"$file_name") or die $!;
         while(my $line=<RF>)
         {
             next if($line=~/^\n/);
             next if($line=~/^\_/);
             chomp($line);
             my @arr=split(/\t/,$line);
             ${$hash{$arr[0]}}{$entity_submitter_id}=$arr[1];
         }
         close(RF);
      }
}
#print Dumper $obj
open(WF,">mRNAmatrix.txt") or die $!;
my $normalCount=$#normalSamples+1;
my $tumorCount=$#tumorSamples+1;
print "normal count: $normalCount\n";
print "tumor count: $tumorCount\n";
print WF "id\t" . join("\t",@normalSamples);
print WF "\t" . join("\t",@tumorSamples) . "\n";
foreach my $key(keys %hash)
{
    print WF $key;
    foreach my $normal(@normalSamples)
    {
        print WF "\t" . ${$hash{$key}}{$normal};
    }
    foreach my $tumor(@tumorSamples)
    {
        print WF "\t" . ${$hash{$key}}{$tumor};
    }
    print WF "\n";
}
```

mRNA_merge.pl 主要程式碼

最終得到 113 個正常組織、1109 個癌變組織的 60484 份基因表達資訊。

由於獲取到的各個基因只有其代號，還需轉化成基因名才方便我們進行差異基因分析，ensembl 中有紀錄人體基因組學資訊，我們從 ensembl 的網站（https://asia.ensembl.org/Homo_sapiens/Info/Index）下載該文件。

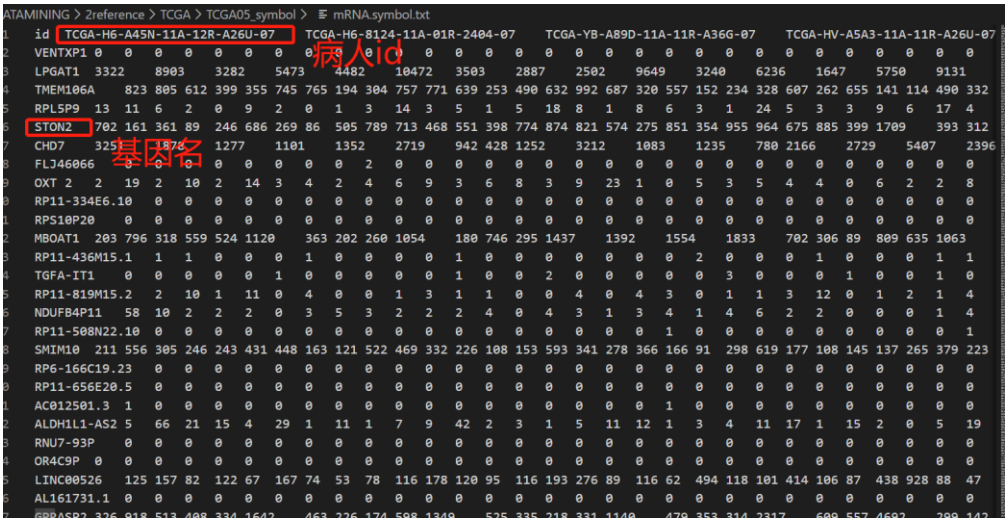


下載流程示意圖

下載完成後將該檔解壓，與 mRNAmatrix.txt 置於同一資料夾，並使用‘perl ensemblToSymbol.pl Homo_sapiens.GRCh38.102.chr.gtf mRNAmatrix.txt mRNA.symbol.txt’執行 perl ensemblToSymbol.pl，該程式碼將所有基因 id 名轉化爲基因實際名，得到轉化爲基因名的檔 mRNA.symbol.txt。

```perl
while(my $line=<RF>)
{
    chomp($line);
    if($line=~/gene_id \"(.+?)\"\;.+gene_name "(.+?)"\;.+gene_biotype \"(.+?)\"\;/)
    {
        $hash{$1}=$2;
    }
}
close(RF);
open(RF,"$expFile") or die $!;
open(WF,">$outFile") or die $!;
while(my $line=<RF>)
{
    if($.==1)
    {
        print WF $line;
        next;
    }
    chomp($line);
    my @arr=split(/\t/,$line);
    $arr[0]=~s/(.+)\..+/$1/g;
    if(exists $hash{$arr[0]})
    {
        $arr[0]=$hash{$arr[0]};
        print WF join("\t",@arr) . "\n";
    }
}
```

ensemblToSymbol.pl 部分 10715002 程式碼



mRNA.symbol.txt 主要內容

3.2 差異基因分析

得到 mRNA.symbol.txt 後我們可以開始對基因進行差異基因表達分析，以篩選出差異基因，由於 edgeR 和 DEeq 各自有其優點，我們分別將所有基因用 edgeR 和 DEeq 進行差異分析後，對其得到的結果取 Intersection。

edgeR 進行差異基因分析主要程式碼如下：

```r
foldChange=1  #fold Change基因差異倍數，增大嚴格
padj=0.05 #P-value adjust減少嚴格

# setwd("D:\\2020\\資料科學\\final-project\\tcga_files\\rename\\edgeR")
setwd("/home/kyro_zhang/ZQX/rename")
library("edgeR")
rt=read.table("mRNA.symbol.txt",sep="\t",header=T,check.names=F)  #
rt=as.matrix(rt)
rownames(rt)=rt[,1]   #第一列基因名為rowname
exp=rt[,2:ncol(rt)]  #第二列到最後一列為表達量
dimnames=list(rownames(exp),colnames(exp))
data=matrix(as.numeric(as.matrix(exp)),nrow=nrow(exp),dimnames=dimnames)
data=avereps(data)
data=data[rowMeans(data)>1,]

#group=c("normal","tumor","tumor","normal","tumor")
group=c(rep("normal",113),rep("tumor",1109))
design <- model.matrix(~group)
y <- DGEList(counts=data,group=group)  #轉化為edgeR對象格式
y <- calcNormFactors(y)       #校正因子
y <- estimateCommonDisp(y)    #估計normal的變異（先估計內部差異程度，再看他們之間的差異是否大於內部差異，如果更大，達到一定水平就可以篩選出
y <- estimateTagwiseDisp(y)   #估計tumor的變異
et <- exactTest(y,pair = c("normal","tumor"))     #檢驗
topTags(et)
ordered_tags <- topTags(et, n=100000)  #顯示前10w，篩選之後結果小於10w，即所有基因都顯示

allDiff=ordered_tags$table
allDiff=allDiff[is.na(allDiff$FDR)==FALSE,]
```

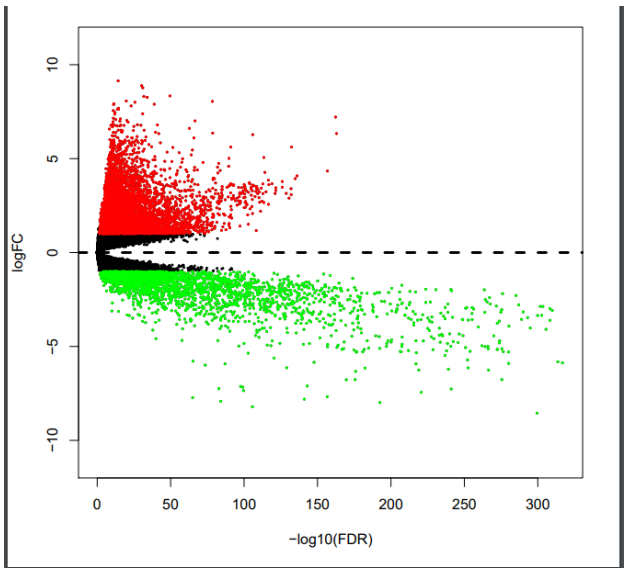
edgeR.R 部分程式碼

其中 fold change 是指基因的差異倍數，logFC 的值是 fold change 取以 2 爲底的對數的值，logFC 的絕對值越大，差異越顯著。pval，即 P-value，是在以‘基因在正常組織樣本中和癌症樣本中無差異‘爲假設時，該假設爲真時所得到的樣本觀察結果或更極端結果出現的概率，padj 是 pval 矯正後的值，我們在該分析中取 fold change=1，padj=0.05，最終得到 normalize 後的差異基因檔 diffmRNAExp.txt 及其分析結果檔 diffSig.xls。

id	logFC	logCPM	PValue	FDR
CKM	-8.314478465	5.269065214	0	0
ACTA1	-6.99003588	6.347895277	0	0
MYLPF	-6.960770696	2.428542234	0	0
PYGM	-6.946510003	3.900152211	0	0
SLN	-6.461278001	1.779353508	0	0
TNNC2	-6.342682634	2.982459099	0	0
ACTN3	-6.31714691	0.925098019	0	0
KLHL41	-6.301494802	3.109751794	0	0
ATP2A1	-6.174821083	3.857630739	0	0
TNNC1	-6.145576356	2.893498448	0	0
NEB	-6.100399337	5.081213366	0	0
MYL3	-6.070565218	1.614761413	0	0
APOBEC2	-6.06851375	0.306638251	0	0
MYOT	-5.886187587	2.082004484	0	0
LDB3	-5.526011287	3.24327476	0	0
TNNI1	-5.427363459	3.527351738	0	0
UNC45B	-5.351456975	-0.199392205	0	0
ACVR1C	-5.179367828	3.232595799	0	0
KCNIP2	-5.121955466	4.01915222	0	0
HSD17B13	-5.074201906	0.197026629	0	0
HSPB7	-5.072304931	4.022561238	0	0

9042	LINC02041	1.33528308017661	0.0412910070099664	0	6.64530786713444	0.945914143161812	4.43824955478669	2.1526005274												
9043	RPRML	5.17451742242429	0.0568193028100674	0.702378833456111	0.607967265763384	6.69079130746919	0.83541136335953	0												
9044	TUNAR	1.40180840295663	34.1234178199219	231.26332824361	9.58624518771664	10.4659134589763	6.91344880962337	0	1											
9045	AC090774.2	2.60748664231719	5.08202335183005	1.49904475510169	1.33493999190142	0.943290199462383	1.7259438340053	0												
9046	C8B	4.01332683636757	1.40647692383941	13.8213527101453	4.06667532898102	5.70666925599133	0	1.32275859192856	4											
9047	AC011840.1	0.0470301702930589	5.07128819971438	0	0.62123377273408	0	0.842627393316794	0	0	0	0	0.05316484569								
9048	C10orf99	0.0476905000528223	0.0479656705884225	3.90097413090873	0	6.71368255262323	0	0	0	0	0	7.7253865196	7							
9049	CDH19	272.08556364122	45.9907614879344	148.549931540285	46.2762074133989	73.3602217752528	183.550300570688	64.32												
9050	CHIAP2	0.0352636908209706	0.0354610435513797	0	3.68076183537608	0	0	0	0	0	0	0.0397170970398768	0.73683460002							
9051	AC011473.3	0.0430328653724596	0.0432783535048673	0.719847898889135	1.35341746903285	0.945106110005698	0	2.13999145965												
9052	AC005303.1	0.0229764555990511	1.26719727741533	0	0	0	0	0	0	0	2.57376888448158	8.33654392068717	0	0	0					
9053	SPRR2A	1.4048911550385	5.34160734766269	27.6351594688919	0	36.2388239127903	0	0	0	0	0	92.3549415306195	0							
9054	AC068756.1	4.995123519551	0.0416653803446624	0	0	2.87062356535558	3.53484910710496	0.651246729706961	0	0	0.780									
9055	CRH	0.0433160986447233	8.65892834605897	0	0	0	2.62925121306356	0	0.944435565417295	0	3.31208842311204	0.048								
9056	RIPPLY2	0.059380467742026	2.67304065766233	32.3958960618589	4.14952955223312	1.89510020331739	1.7082831381545	3.515												
9057	LINC02866	8.70086855833442	0.0459510639527179	4.72817714239477	1.34299178498112	2.86542492522909	0	3.63294361296												
9058	CDH10	10.5056105413774	10.5210949360198	3.75579299035342	4.05947735057446	7.61324930915957	5.18221823433009	1												
9059	PAGE4	0.0408261719687237	1.33648455887672	5.60410684338539	0	0	0.848808192284779	2.15412420010055	0.9454634840											
9060	NR0B1	18.3312894782175	1.40741954247647	13.748591981767	13.1031096079187	33.374750926388	10.4078471663298	5.5704228												
9061	SLCO6A1	1.31074413265327	8.45366734288663	0	0	0	0	0	0	0	3.80535161833799	0	0	0.0838001623118497	0					
9062	LINC01511	1.40356507598424	0.0778766257961987	598.829564041305	3.35774053185247	0	2.56657518764185	0	44.780234											
9063	PGBD4P8	0.0497016178067192	0.0499897974391342	56.6725926962642	0	0	0	0	0	0	0	0.0562321795727786	0.71267879562							
9064	SPRR3	0.079475500410279	0.0799701020486019	100.409914652347	0.604848714600905	148.700915998438	0	0	0.93866326513											
9065	ADGRG7	0.0774222583166004	0.077901781453761	4.51459277315009	0	0	1.69831129954747	0	2.84154647515277	1.9894480										
9066																				

edgeR 分析得到 9064 個基因



edgeR 分析後的基因 heatmap，紅色和綠色爲我們篩選出的基因

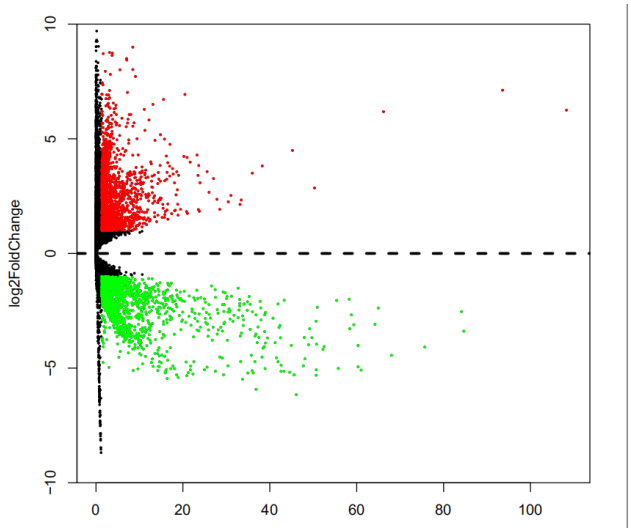
我們使用 DESeq 輸入同樣參數對基因進行差異分析，得到 3618 個差異基因。

id	baseMean	baseMeanA	baseMeanB	foldChange	log2FoldChange	pval	padj	
MMP11	17170.5467722107	248.439329691569	18894.8011824944	76.0539855181216	6.24895194752574	1.6973522284754e-113		
COL10A1	8580.1358680035	9	67.9338182876425	9447.4747603551	7	139.068802527087	7.11965500442496	1.7117565757673e-98 2
CAV1	8624.91456747784	48164.9228538502	4596.04086471852	0.0954229881913145	-3.38951932444088	2.27090235623733e-89		
GSN	37594.7650199514	150996.854033945	26039.8181682099	0.172452719858362	-2.53572721227457	9.87632395031372e-89	7	
OXTR	2017.28731363431	13819.1465446439	814.753415434057	0.0589583020052598	-4.08416121425399	3.63304443044241e-80		
CHRDL1	2831.43935655582	21094.0830483169	970.592884807396	0.0460125658263604	-4.44182828121981	2.00294570892168e-72		
COL11A1	12009.3896256926	181.96683766383	13214.528286691	72.62053051174	6.18230556432403	1.5385414689485e-70	6.65155435354	
DST	16167.1124445592	60526.9937197586	11647.1245418562	0.19242859798692	-2.37760487214145	2.89711728379569e-69	1	
SYNM	5994.18090094062	30115.3981078632	3536.38329554634	0.117427745197995	-3.0901547739147	1.82093506458985e-68		
SCARA5	825.885634147081	6927.59381808007	204.160634341465	0.0294706415680195	-5.08457772188407	3.16362157847019e-65		
ITIH5	3009.40815541892	20238.5943759992	1253.86438361949	0.0619541239042981	-4.01265587289289	1.82967694928966e-64		
LYVE1	765.592989462469	6277.87884837956	203.926350997518	0.0324833205486523	-4.94415707282389	1.9117105649558e-64	4	
TNS1	15244.7911229931	77096.7575307895	8942.47173247827	0.115990244192915	-3.10792462786543	1.97372716630198e-63		
EGR1	21871.5395987728	93347.3192312538	14588.6152538942	0.156283173143443	-2.67776564175356	7.41682991474377e-63		
MME	3249.03126125423	17457.5123209808	1801.27800629562	0.103180680796702	-3.27675522420306	1.88140229391649e-62	3	
DCN	46239.7053408068	144297.586607032	36248.2350224267	0.251204721262193	-1.99306451569945	2.67526285407452e-62	5	
CD300LG	621.629826923729	5158.1222546713	159.390291722282	0.0309008363503681	-5.01621030367745	9.51300822915816e-60		
TXNIP	42945.6937913307	137816.526098979	33278.9633578192	0.241472951755572	-2.0500664981805	2.4504720134677e-59	4	
TNXB	2497.25941783761	17016.6981726197	1017.82156455503	0.0598131055878237	-4.0633945632941	1.90009515108579e-56		
CD36	13200.9644739294	92720.9077948317	5098.39134925681	0.0549864261525381	-4.18478066821601	3.93049286394881e-56		
ANXA1	12023.2199326936	44348.6706238057	8729.46346010958	0.196837094265093	-2.3449259704022	8.3702405851045e-55	1	
CAVIN2	1327.54427082065	8793.88460492278	566.771991511772	0.0644506969302844	-3.95566023068463	1.29239841301783e-54		

分析結果

3606	NLRP13	7.93709687413347	20.4240598775647	6.66475528767024	0.326318828265447	-1.61564586301529	0.00773426321936541 0
3607	DNAJC18	549.806822459133	1008.03919084865	503.115877799065	0.499103489592999	-1.0025891040896	0.00773702875634822 0
3608	AC119150.1	4.16218144059769	0.370364141890527	4.54854334749932	12.2812735711434	3.61838827119003	0.007754140871422
3609	ADAMTS19	131.049827818367	29.1393530820178	141.433852746417	4.85370599506195	2.27908672235138	0.00778406581078 2
3610	RHPN1-AS1	297.812485558105	91.0860561365267	318.876585219636	3.50082766501252	1.80769604407111	0.00778422500126 5
3611	CCND2-AS1	2.2148123627567	6.27425610052793	1.80118103510282	0.28707483504718	-1.80050122531006	0.00779046365975604 0
3612	JPH3	90.7427470030828	18.3108466329629	98.123093929885	5.35874150970417	2.4218942262634	0.00779496672153404 0.0494651
3613	HLA-DQB2	1078.69980694946	512.13363371769	1136.42927275216	2.21900925448416	1.1499156843761	0.00781370311226657 0.049
3614	GRPR	678.106707794919	289.020847747063	717.752065942266	2.48339201665621	1.31231201658731	0.00782051601454085 0
3615	BARX1-DT	2.72837343275427	0.242116194207921	2.98170712793528	12.315190802044	3.62236707489884	0.00783274624419718 0
3616	PASK	1074.86427242543	517.744502406496	1131.63121021816	2.18569430473582	1.1280916371116	0.00786338924813032 0.049
3617	DSCR8	34.6935420052176	0.518814764384298	38.1757279188463	73.5825780982759	6.20129231950538	0.00788209631024597 0
3618	TTC7B	930.859739992209	1707.12340810937	851.763441978467	0.498946612724261	-1.0030426394019	0.00788312289638604 0
3619	GRAMD4P7	29.2486793563566	3.64751157321369	31.8572744505812	8.73397487880017	3.12663838155525	0.007896463325714
3620							

DESeq 得到 3618 個基因



DESeq 分析後的基因 heatmap，紅色和綠色爲我們篩選出的基因

3.3 資料前處理

在資料前處理部分，我們首先執行 clinical.ipynb，觀察從 metadata 中提取出來的 clinical 資料集並進行初步處理，再執行 filter.ipynb 對 clinical 資料集進行進一步處理。由於資料有大量缺失值及無用資訊，我們在 clinical.ipynb 刪除掉無用資訊，最終保留了年齡、性別、種族等有效資訊。

```
clinical_df.isnull().sum()
```

```
case_id                                0
case_submitter_id                      0
project_id                             0
age_at_index                           0
age_is_obfuscated                   2194
cause_of_death                      2194
cause_of_death_source               2194
country_of_residence_at_enrollment  2194
days_to_birth                         30
days_to_death                       1892
ethnicity                              0
gender                                 0
occupation_duration_years           2194
premature_at_birth                  2194
race                                   0
vital_status                           0
weeks_gestation_at_birth            2194
year_of_birth                          6
year_of_death                       1986
age_at_diagnosis                      30
ajcc_clinical_m                     2194
ajcc_clinical_n                     2194
ajcc_clinical_stage                 2194
ajcc_clinical_t                     2194
ajcc_pathologic_m                      0
```

進行資料篩選前

```
clinical_df.isnull().sum()
```

```
case_submitter_id           0
age_at_index                0
ethnicity                   0
gender                      0
race                        0
vital_status                0
ajcc_pathologic_m           0
ajcc_pathologic_n           0
ajcc_pathologic_stage       0
ajcc_pathologic_t           0
primary_diagnosis           0
prior_malignancy            0
tissue_or_organ_of_origin   0
treatment_type              0
futime                      0
dtype: int64
```

刪除無效資料後的 attribute

隨後執行 DNA_preprosess.ipynb，將臨床資訊**轉換為數字**之後和病人的差異基因資訊合併。

1. 在我們分別使用 DESeq 和 edgeR 得到差異基因結果 diffSig_D.txt 和 diffSig.xls 後，我們將其進行 itersection，得到 3562 個差異基因。

```
edge=pd.DataFrame(edge_df['id'])
itersection=pd.merge(DES_df,edge,how='inner')
itersection.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3563 entries, 0 to 3562
Data columns (total 8 columns):
id              3563 non-null object
baseMean        3563 non-null float64
baseMeanA       3563 non-null float64
baseMeanB       3563 non-null float64
foldChange      3563 non-null float64
log2FoldChange  3563 non-null float64
pval            3563 non-null float64
padj            3563 non-null float64
dtypes: float64(7), object(1)
memory usage: 250.5+ KB
```

2. 根據這 3562 個差異基因的基因名，從病人所有基因的表達量中，篩選出這 3562 個基因的表達量，其餘的基因資訊則刪除。

```
des_f = pd.DataFrame(itersection['id'])
exp_f=pd.merge(des_f,exp,how='inner')
```

```
exp_f.head(15)
```

	id	TCGA-BH-A1FH-11B-42R-A13Q-07	TCGA-A7-A13E-11A-61R-A12P-07	TCGA-BH-A0DK-11A-13R-A089-07	TCGA-BH-A18Q-11A-34R-A12D-07	TCGA-BH-A0BJ-11A-23R-A089-07	TCGA-BH-A1FD-11B-21R-A13Q-07	TCGA-BH-A1FN-11A-34R-A13Q-07	TCGA-AC-A2FM-11B-32R-A19W-07	TCGA-E9-A1NA-11A-33R-A144-07	...	11I
0	MMP11	283.354927	281.298283	282.684162	131.750821	270.574934	95.295040	535.179290	247.989614	148.209298	...	116
1	COL10A1	25.636874	51.639298	158.150328	23.819357	30.612493	47.647520	38.720982	52.573798	17.376263	...	71
2	CAV1	41731.434147	114718.060052	9453.875192	25014.047047	13529.734219	82249.149080	30257.681800	16398.065268	43533.670429	...	50
3	GSN	150941.820108	139587.817888	57750.846284	78848.027841	63343.172138	142900.976808	49061.558777	174826.726462	212137.589996	...	160
4	OXTR	7664.076111	5931.724657	20630.595751	8463.315427	12915.509368	7430.414169	14295.648337	16048.895891	6957.659942	...	3
5	CHRDL1	31963.785033	47302.956151	2224.036745	12348.103748	9250.897756	25067.793448	11365.991166	3814.080271	35837.008263	...	16

刪除後僅保留差異基因的表達量

3. 將行列的內容轉置變成病人名字爲 id。

```
transfer=exp_f.values
index1=list(exp_f.keys())
transfer=list(map(list, zip(*transfer)))
transfer = pd.DataFrame(transfer, index=index1)
transfer.head()
```

	0	1	2	3	4	5	6	7	8	9	...	3553	3554	3555	3556	3557	35
id	MMP11	COL10A1	CAV1	GSN	OXTR	CHRDL1	COL11A1	DST	SYNM	SCARA5	...	ADAMTS19	RHPN1-AS1	CCND2-AS1	JPH3	HLA-DQB2	GR
TCGA-BH-A1FH-11B-42R-A13Q-07	283.355	25.6369	41731.4	150942	7664.08	31963.8	28.3355	56722.3	33726	7778.77	...	35.082	82.3079	9.44516	18.8903	717.832	32.38
TCGA-A7-A13E-11A-61R-A12P-07	281.298	51.6393	114718	139588	5931.72	47303	154.918	39057	19067.1	12125.7	...	29.8964	91.0482	0	16.3071	554.443	332.9
TCGA-BH-A0DK-11A-	282.684	158.15	9453.88	57750.8	20630.6	2224.04	77.1651	66539.3	47910.4	3885	...	51.1888	139.814	3.82006	3.05604	407.982	640.2

4. 使用對病人的臨床資訊進行 OrdinalEncoder，並將生存時間 ftime 由按日期計算改爲按年計算，將生存狀態由 Alive/Dead 改爲 1/0。

```
oe = OrdinalEncoder()
clinical_df.loc[:,cate] = oe.fit_transform(clinical_df.loc[:,cate])
clinical_df.head()
```

	id	ethnicity	gender	race	age_at_index	age_at_diagnosis	ajcc_pathologic_m	ajcc_pathologic_n	ajcc_pathologic_stage	ajcc_pathologic_t	primary_diag
0	TCGA-E9-A295	1.0	0.0	3.0	71	25957.0	0.0	2.0	4.0	4.0	
1	TCGA-AO-A0J9	1.0	0.0	3.0	61	22642.0	0.0	11.0	9.0	4.0	
2	TCGA-E2-A1L7	1.0	0.0	3.0	40	14854.0	0.0	9.0	7.0	4.0	

```
# format(float(futime)/365,'.4f')
clinical_df['futime']=clinical_df['futime'].apply(lambda x: x/365)
clinical_df['age_at_diagnosis']=clinical_df['age_at_diagnosis'].apply(lambda x: x/365)
clinical_df.head()
```

	id	ethnicity	gender	race	age_at_index	age_at_diagnosis	ajcc_pathologic_m	ajcc_pathologic_n	ajcc_pathologic_stage	ajcc_pathologic_t	primary_diag
0	TCGA-E9-A295	1.0	0.0	3.0	71	71.115068	0.0	2.0	4.0	4.0	
1	TCGA-AO-A0J9	1.0	0.0	3.0	61	62.032877	0.0	11.0	9.0	4.0	
2	TCGA-E2-A1L7	1.0	0.0	3.0	40	40.695890	0.0	9.0	7.0	4.0	

```
clinical_df['fstatus'] = clinical_df['fstatus'].map({'Alive': 1, 'Dead': 0})
clinical_df.head()
```

	id	ethnicity	gender	race	age_at_index	age_at_diagnosis	ajcc_pathologic_m	ajcc_pathologic_n	ajcc_pathologic_stage	ajcc_pathologic_t	primary_diag
0	TCGA-E9-A295	1.0	0.0	3.0	71	25957.0	0.0	2.0	4.0	4.0	
1	TCGA-AO-A0J9	1.0	0.0	3.0	61	22642.0	0.0	11.0	9.0	4.0	

5. 根據病人 id 將基因表達量和臨床資訊合併到一起。

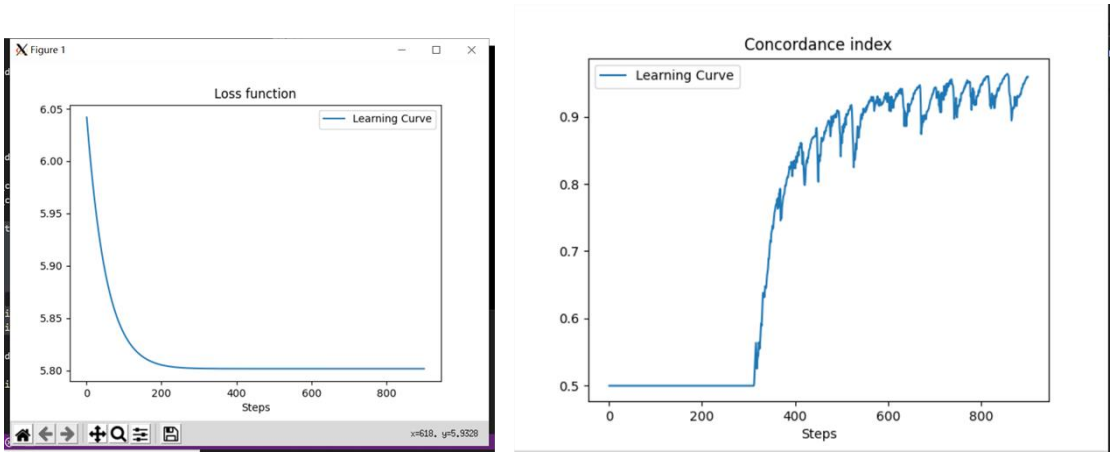
```
clinical_gene10=pd.merge(filter10,clinical_df,how='inner')
clinical_gene10.head()
```

id	MMP11	COL10A1	CAV1	GSN	OXTR	CHRDL1	COL11A1	DST	SYNM	...	ajcc_pathologic_m	ajcc_pa
CGA-3C-AAAU	21873.126656	3520.656342	1012.464785	9876.390778	225.286667	236.771869	1597.326642	12505.618701	2793.554669	...	2.0	
CGA-3C-AALI	18143.286550	7426.918006	2303.185472	22327.786889	3922.256448	380.538689	15369.772361	11665.577409	1878.464389	...	0.0	
CGA-3C-AALJ	27974.425254	7941.952116	3618.158269	25587.577390	177.592847	776.672717	8676.002550	6158.919933	596.711966	...	0.0	
CGA-3C-AALK	16301.894223	14298.570862	4169.872416	33188.136433	938.003659	369.978241	19877.625110	11133.080555	3103.464543	...	0.0	

3.4 訓練模型

在準備好資料集後，執行 training.py 訓練 model，在 train model 之前需要先將 Deepsurv 從 github 中下載下來，我們在訓練模型的過程中將資料集按 training：testing=4：1 的比例進行訓練。

在 train 該 model 的過程中，我們先將 learning rate，L1regulation 等取相對大的範圍值，通過觀察 loss function 和 C-index 值，不斷縮小範圍。



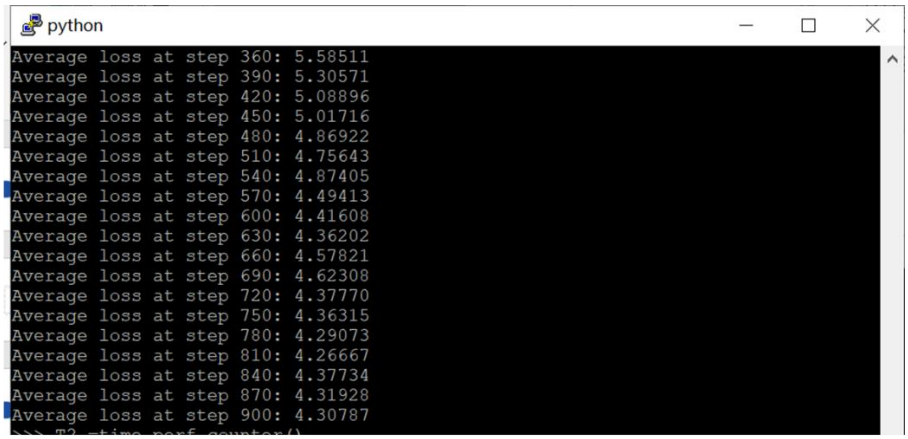
Train model 過程中的 loss function 及 C-index

四．實作結果

最終我們在選取 1000 個基因和臨床資訊進行生存分析時，達到了 C-index 平均值爲 0.95 的值。

```
hidden_layers_nodes = [5,20,110,1]
nn_config = {
    #-------0.08 0.03 0.001
    "learning_rate": 0.001,
    "learning_rate_decay": 1.0,
    "activation": 'relu',
    "L1_reg": 5e-06,
    "L2_reg": 0.05,
    "optimizer": 'adam',
    "dropout_keep_prob": 1.0,
    "seed": 1
}
```

最終 neural network 的參數




```
CI: 0.9599261484541244
```

最終結果

五．總結與感悟

我們在本次實作中，利用在課堂上及課外所學到的知識，使用 edgeR、DESeq 和 Deepsurv 成功地將原本不知如何下手的 final project 進行實踐，並且最終得到了一個相對較好的結果。儘管本次實作結果十分簡單，但實作過程並不容易，報告中的僅爲實作的一部分，我們在做生存分析的 research 時也花費了很多時間。在實作過程中，我們發現資料的獲取和前處理相比訓練 model 更爲複雜，尤其是面對雜訊較多的原始資料集，它們不像 kaggle 上已經處理完成的資料集，需要我們進行進一步處理，資料探勘的過程正是像這樣一步步分析原始資料並獲得結果的過程。

參考資料

[1] Mark D. Robinson, Davis J. McCarthy and Gordon K. Smyth, “edgeR: a Bioconductor package for differential expression analysis of digital gene expression data”, *Bioinformatics*, vol. 26, Issue 1, pp.139–140, 1 January 2010.

[2] Yunshun Chen, Aaron T. L. Lun and Gordon K. Smyth, “Differential Expression Analysis of Complex RNA-seq Experiments Using edgeR”, in *Statistical Analysis of Next Generation Sequencing Data*, ch.3, pp.51-74, June 2014.

[3] Simon Anders, Wolfgang Huber, “Differential expression analysis for sequence count data”, *nature precedings*, April 2010.

[4] D. R. Cox, “Regression Models and Life-Tables”, *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 34, no. 2 (1972), pp. 187-220, January 1972.

[5] Knut Liestbl, Per Kragh Andersen, and Ulrich Andersen, “Survival analysis and neural nets”, *Statistics in medicine*, vol.13, Issue12, pp.1189-1200, June 199a4.

[6] W Nick Street, “A neural network model for prognostic prediction”, in *ICML*, pp. 540–546, 1998.

[7] Leonardo Franco, Jose M Jerez, and Emilio Alba, “Artifificial neural networks and prognosis in medicine. survival analysis in breast cancer patients”, in *ESANN*, pp.91–102. i6doc, January 2005.

[8] Elia Biganzoli, Patrizia Boracchi, Luigi Mariani, and Ettore Marubini, “Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach”, *Statistics in medicine*, vol.17, Issue10, pp.1169-1186, May 1998.

[9] David Faraggi and Richard Simon, “A neural network model for survival data”, *Statistics in medicine*, vol.14, Issue1, pp.73-82, January 1995.

[10] Jared L. Katzman, Uri Shaham, Alexander Cloninger, and et al., “DeepSurv: Personalized Treatment Recommender System Using A Cox Proportional Hazards Deep Neural Network”, *BMC Medical Research Methodology*, Feb 2018.