LAMPIRAN

Lampiran 1. *Script* pemanggilan dan *preprocessiong*

```
## calling data
files = list.celfiles("E:/INES/data1/10072", full.names
+= TRUE)
dataines = ReadAffy(filenames = files)

## pheno data
p1<-read.AnnotatedDataFrame(file.path("E:/INES/script
+/parupdata.txt"), sep = "\t", header = TRUE)
phenoData(dataines) <- p1
pheno =   pData(phenoData(dataines))
table(pheno$statusstage)

## PREPROCESSING DATA
rma_data<-  threestep(dataines,  background.method  =
+"RMA.2",normalize.method = "quantile",
+summary.method="median.polish")
```

Lampiran 2. Pembuatan *Expression set*

```
## EXPRESSION SET######
exp = exprs(rma_data)
as(rma_data,"ExpressionSet")
all(rownames(pData)==colnames(exp))
experimentData <- new("MIAME",
                name="Gene Expression data Lung Cancer",
                lab="Lab of Cancer Biology and Genetics",
                 contact="240-435-8956",
                 title="Gene  expression  signature  of
                 cigarette smoking and its role in lung
```

```
                adenocarcinoma development and
                surviva",
                abstract="Tobacco smoking is
                responsible for over 90% of lung cancer
                cases, and yet the precise molecular
                alterations induced by smoking in lung
                that develop into cancer and impact
                survival have remained obscure",
                url="submission data : 20 february
                2008",
          other=list(notes="type : 18 Agustus 2014"))

ines <- ExpressionSet(assayData=exp,
                      phenoData=p1,
                      experimentData=experimentData,
                      annotation="hgu133a")
```

Lampiran 3.*Filtering* dan *feature selection*

```
## filtering
filters = nsFilter(ines, require.entrez=TRUE,
                   remove.dupEntrez=TRUE,
                   var.cutoff=0.5,
                   feature.exclude="^AFFX")
filters$filter.log
filter=filters$eset

##feature selection
f<- ttest(filter$statusstage,p=0.05)
ff <- filterfun(f)
selGenes <- genefilter(exprs(filter),ff)

sum(selGenes)
```

```
inesfit<- filter[selGenes,]
inesfit

## plot filtering
exp = exprs(inesfit)
plot(rowIQRs(exp),rowMedians(exp),
     xlab='IQR expression level',
     ylab='Median expression level',
     main='Distribution Properties of the Selected
     Genes',col=c("red","blue"))
```

Lampiran 4. Penggabungan data fenotip dan gene expression dan *Split data*

```
## Kombinasi data
data   = as.data.frame(t(exprs(inesfit)))

gender  = as.factor(inesfit$Gender)
gender = as.numeric(gender)

smoking = as.factor(inesfit$smoking.s)
smoking = as.numeric(smoking)

tissue = as.factor(inesfit$tissue)
tissue = as.numeric(tissue)

statusstage = as.factor(inesfit$statusstage)
statusstage = as.numeric(statusstage)

ines1=as.data.frame(cbind(data,gender,smoking,
+tissue,statusstage))
head(ines1)
```

```
##### split data #####
early = which(ines1$statusstage=="1")
late = which(ines1$statusstage=="2")
spliter = sample.split(early, SplitRatio = 0.7)
splitla = sample.split(late, SplitRatio = 0.7)

traininger = subset(early,spliter==TRUE)
tester = subset(early,spliter==FALSE)

trainingla = subset(late,splitla==TRUE)
testla = subset(late,splitla==FALSE)

training2 = c(traininger,trainingla)
testing2 = c(tester,testla)

train = ines1[training2,]
test  = ines1[-training2,]

## Penyimpanan data training dan tetsing
#####training training
save(train, file="train.rda")
save(test,file="test.rda")

######
load(file = "train.rda")
load(file="test.rda")
```

Lampiran 5. Deskripsi data, analisis KNN dan hasil akurasi

```
# Deskripsi data
p <- ggplot(gender, aes(x=Gender, y=value, fill=Gender))
    +
  geom_bar(stat="identity", position=position_dodge())+
  geom_text(aes(label=value), vjust=1.6, color="black",
            position   =   position_dodge(0.9),   size=
    4)+theme_minimal()


p  <-  ggplot(status,  aes(x=Status.merokok,  y=Value,
    fill=Status.merokok)) +
  geom_bar(stat="identity", position=position_dodge())+
  geom_text(aes(label=Value), vjust=1.6, color="black",
            position   =   position_dodge(0.9),   size=
    4)+theme_minimal()

p+scale_fill_manual(values=c("#999999",      "#E69F00",
    "#56B4E9"))

p    <-    ggplot(jaringan,    aes(Jaringan,    y=value,
    fill=Jaringan)) +
  geom_bar(stat="identity", position=position_dodge())+
  geom_text(aes(label=value), vjust=1.6, color="black",
            position   =   position_dodge(0.9),   size=
    4)+theme_minimal()
# using Cross Validation  ##
trctrl <- trainControl(method = "repeatedcv", repeats
          =3)
set.seed(3333)
fa = factor(train$statusstage)
```

```
knnfit <- train(train,fa,method = "knn",trControl =
    trctrl,preProcess=c("center","scale"),tuneLength =
    20)
#################################################
plot(knnfit,
     xlab="Number of neighbourn(k)",
     main="Corparison of Accuracy against k",
     type="b",
     col="blue",
     lwd=1.8,pch="O")

testperd <- predict(knnfit,newdata = test)
testperd
View(data.frame(testperd,test$statusstage))

confusionMatrix(data= testperd,
                reference=test$statusstage,
                dnn = c("predict values","Actual
               values"))

#########with ROC #########

predict <- as.matrix(predict(knnfit, test, type="prob"))
pred <- predict[,2]
pred <- prediction(pred,test$statusstage)
predi <- performance(pred,"tpr","fpr")

plot(predi,colorize = F,main="ROC curve",ylab
    ="Sensitivity",xlab="1-specifisity")
abline(a=0,b=1)
auc <- performance(pred,"auc")
auc <- unlist(slot(auc,"y.values"))
auc <- round(auc,4)
```

```
legend(0.6,0.6,auc,title="AUC",cex = 0.6)
```

Lampiran 6. Gen yang paling berpengaruh dan annotasinya.

```
#####################gene imp#################
knnfit
knnImp <- varImp(knnfit)
knnImp
plot(knnImp, top=50)
plot(varImp(knncv))
```

Lampiran 7. Hasil *remove filteringi* , struktur data, dan hasil nilai k terbaik

##remove filter ##

```
> filters$filter.log
$numDupsRemoved
[1] 7407

$numLowVar
[1] 6218

$numRemoved.ENTREZID
[1] 2431

$feature.exclude
[1] 10
```

##struktur data ##

```
> ines
ExpressionSet (storageMode: lockedEnvironment)
assayData: 22283 features, 107 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: GSM254625 GSM254626 ... GSM254731 (107 total)
  varLabels: organism_ch1 Gender ... statusstage (8 total)
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu133a
```

##k terbaik ##

```
145 predictors
  2 classes: '1', '2'

Pre-processing: centered (145), scaled (145)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 67, 67, 66, 66, 67, 67, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.8180556  0.32013072
   7  0.8335317  0.36098039
   9  0.8611111  0.43764706
  11  0.8295635  0.30392157
  13  0.8343254  0.32352941
  15  0.8295635  0.30980392
  17  0.8343254  0.32352941
  19  0.8164683  0.25686275
  21  0.8123016  0.23686275
  23  0.8075397  0.21725490
  25  0.7992063  0.17725490
  27  0.7807540  0.09843137
  29  0.7617063  0.02000000
  31  0.7575397  0.00000000
  33  0.7575397  0.00000000
  35  0.7575397  0.00000000
  37  0.7575397  0.00000000
  39  0.7575397  0.00000000
  41  0.7575397  0.00000000
  43  0.7575397  0.00000000

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was k = 9.
```

Lampiran 8. Hasil klasifikasi KNN dan ROC

	testperd	test.statusstage
1	1	1
2	1	1
3	2	2
4	1	1
5	1	2
6	1	1
7	1	1
8	1	1
9	1	2
10	1	2
11	1	2
12	1	1
13	2	2
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	1	2
20	2	2
21	1	1
22	1	1
23	1	1
24	1	1
25	1	1
26	1	1
27	1	1
28	1	1
29	1	1
30	2	2
31	1	1
32	1	1
33	1	1