weekly_july3

Dohun Lee

2022-07-14

This is a RMDmarkdown to track and show progress between 13/07/22 ~ 27/07/22.

These were the main things to tackle:

1. Absolute ratio 0~1
2. ichorCNA: correlation between liquidCNA and ichorCNA's optimichorCNA & CTC_count
3. Compare the absolute ratios between timely batch 2 options
4. RECIST: look at metastatic data; look at progression; evaluate whether subclonality tells us anything; compare subclonality between Y and N metastasis
5. Investigate something about purity using timely patient. Effect of purity filtering on plot.3C and absolute subclonality measured

1. Priority: fix absolute ratio 0~1

Patient 3080 | patient with extreme absolute sub-clonality estimation:

```
liquidCNA_results$patient_3080
```

```
##      time         relratio              rat           rat_sd purity_mean
## 1 Sample3 0.86965558464975 8.47355873449432 1.58345586022625        0.13
## 2 Sample2                1 9.74634357458251 1.78007901062353        0.07
## 3 Sample1                0                0                0        0.05
##   purity_median seg_used cutOff
## 1         0.245       25    0.3
## 2          0.07       25    0.3
## 3          0.05       25    0.3
```

I think the extreme absolute subclonality ratio is coming from extreme deltaCN values

```
head(seg.dcn.toUse.46)
```

```
##       Sample3    Sample2
## 1  10.5037937  12.635239
## 2  10.6236352  12.513129
## 4 -12.0265727 -13.242563
## 5   0.6779776   1.024608
## 6 -11.8246368 -12.767660
## 7  11.1478483  13.070224
```

The extreme deltaCN seems to come from low purity...

The three time samples have the following purity estimates:

```
pVec.46
```

```
## [1] 0.05 0.07 0.13
```

This results the sample with low purity to have extreme CNS when corrected by purity.

Before correction the segment CN values look normal...

```
head(seg.cns.46)
```

```
##    Sample1  Sample2  Sample3
## 1 1.371152 2.004080 1.730489
## 2 1.366683 1.989275 1.734448
## 3 2.019916 2.063101 2.033835
## 4 2.742167 2.112054 2.366179
## 5 1.952375 2.005047 1.964311
## 6 2.745858 2.150465 2.402029
```

However, with correction...

```
head(seg.cns.corr.46)
```

```
##      Sample1  Sample2     Sample3
## 1 -10.576953 2.058286 -0.07315924
## 2 -10.666340 1.846789 -0.04270458
## 3   2.398312 2.901436  2.26027197
## 4  16.843336 3.600773  4.81676317
## 5   1.047492 2.072100  1.72546964
## 6  16.917166 4.149506  5.09252909
```

The equation they correct CNS with purity by: **(((CNS-2)*1/purity)+2)**

If we correct the CNS value of the first segment for each sample...

```
(1.371152    - 2)/0.05 + 2 #Sample1
```

```
## [1] -10.57696
```

```
(2.004080 - 2)/0.07 + 2 #Sample2
```

```
## [1] 2.058286
```

```
(1.730489 - 2)/0.13 + 2 #Sample3
```

```
## [1] -0.07316154
```

This results dCN to be extreme

```
head(seg.dcn.46)
```

```
##   Sample1     Sample2     Sample3
## 1       0  12.6352387  10.5037937
## 2       0  12.5131288  10.6236352
## 3       0   0.5031239  -0.1380404
## 4       0 -13.2425631 -12.0265727
```

```
## 5       0   1.0246081   0.6779776
## 6       0 -12.7676596 -11.8246368
```

So it seems that the extreme subclonality estimates are coming from low purity and purity-corrected CNS rather than the GMM fitting step

Take the samples where sub-ratio is outside [0,1] and check the corresponding purity: Majority of the error looks to come from small purity values. Currently, 31 out of 283 estimates are outside [0,1]; this is from 22/80 patients. Additionally, only 23 of the estimates linked with RECIST are outside the boundary.

```
rats <- unlist(sapply(1:length(liquidCNA_results), function(x) as.numeric(liquidCNA_results[[x]]$rat)))
purities <- unlist(sapply(1:length(liquidCNA_results), function(x) as.numeric(liquidCNA_results[[x]]$pu

# 31/283 estimations have sub-clonality ratio outside of [0,1]
out.rats<- which(rats < 0 | rats > 1)
length(out.rats)
```

```
## [1] 31
```

```
is.out.rat <- function(x){
  rats <- as.numeric(liquidCNA_results[[x]]$rat)
  return(any(rats < 0 | rats > 1))
}

# 22/80 patients have sub-clonality ratio outside of [0,1]
out.rat.patients <- which(sapply(1:length(liquidCNA_results), function(x) is.out.rat(x)))
length((out.rat.patients))
```

```
## [1] 22
```

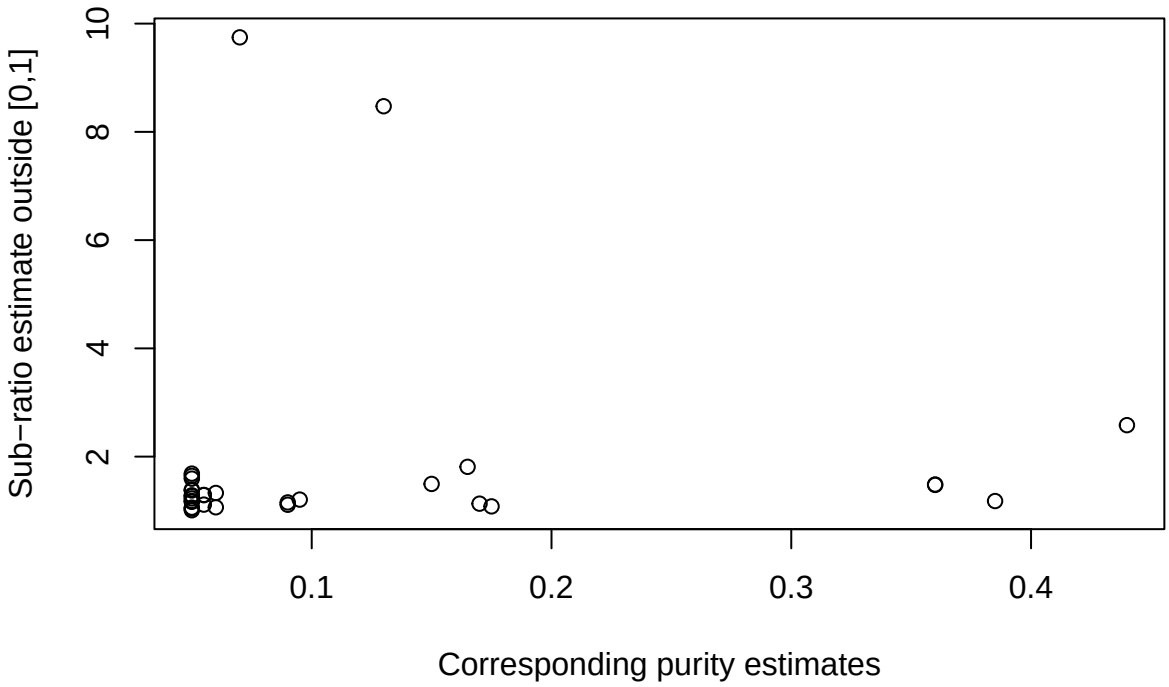
```
#purities of out.rats
rats[out.rats]
```

```
##  [1] 1.110760 1.484909 1.593358 1.389029 1.292059 1.686263 1.164457 1.811724
##  [9] 1.477231 1.206072 1.288886 1.040109 1.268736 1.131115 1.009573 8.473559
## [17] 9.746344 1.154383 1.644420 1.183017 2.581482 1.374887 1.057119 1.227743
## [25] 1.285324 1.081085 1.495673 1.113131 1.064325 1.328851 1.179254
```

```
purities[out.rats]
```

```
##  [1] 0.090 0.360 0.050 0.050 0.055 0.050 0.050 0.165 0.360 0.095 0.055 0.050
## [13] 0.050 0.170 0.050 0.130 0.070 0.090 0.050 0.050 0.440 0.050 0.050 0.050
## [25] 0.050 0.175 0.150 0.055 0.060 0.060 0.385
```

```
plot(purities[out.rats], rats[out.rats],
     ylab = "Sub-ratio estimate outside [0,1]",
     xlab = "Corresponding purity estimates")
```




```
out.results <- sapply(out.rat.patients, function(x) liquidCNA_results[[x]], simplify = F)
names(out.results) <- paste0("patient_", patient_ids[out.rat.patients])
```

Feedback: "Nevertheless, it would be nice to see what causes the high-purity samples in your plot to fail as well? You can focus on the 2-3 with the highest purity and investigate them further."

Investigating why purities are failing:

idea 1: when one of the samples have drastically lower purity

```
out.results$patient_3614
```

```
##      time          relratio               rat            rat_sd purity_mean
## 1 Sample3 0.498681586968614 0.815406233134759  0.11763824770728        0.43
## 2 Sample2                 1  1.49567259622457 0.129331328770106        0.15
## 3 Sample1                 0                 0                 0        0.49
##   purity_median seg_used cutOff
## 1          0.43       65   0.24
## 2          0.15       65   0.24
## 3         0.495       65   0.24
```

Investigating why high purity fails:

```
out.results$patient_3301
```

```
##      time           relratio               rat             rat_sd purity_mean
## 1 Sample3 0.0651646493939343 0.112021769458375 0.0286482270013781       0.195
## 2 Sample2                  1  2.58148214977005    0.5471678581646        0.44
## 3 Sample1                  0                 0                  0       0.175
##   purity_median seg_used cutOff
## 1         0.195       22  0.295
## 2          0.05       22  0.295
## 3         0.175       22  0.295
```

```
pVec.58
```

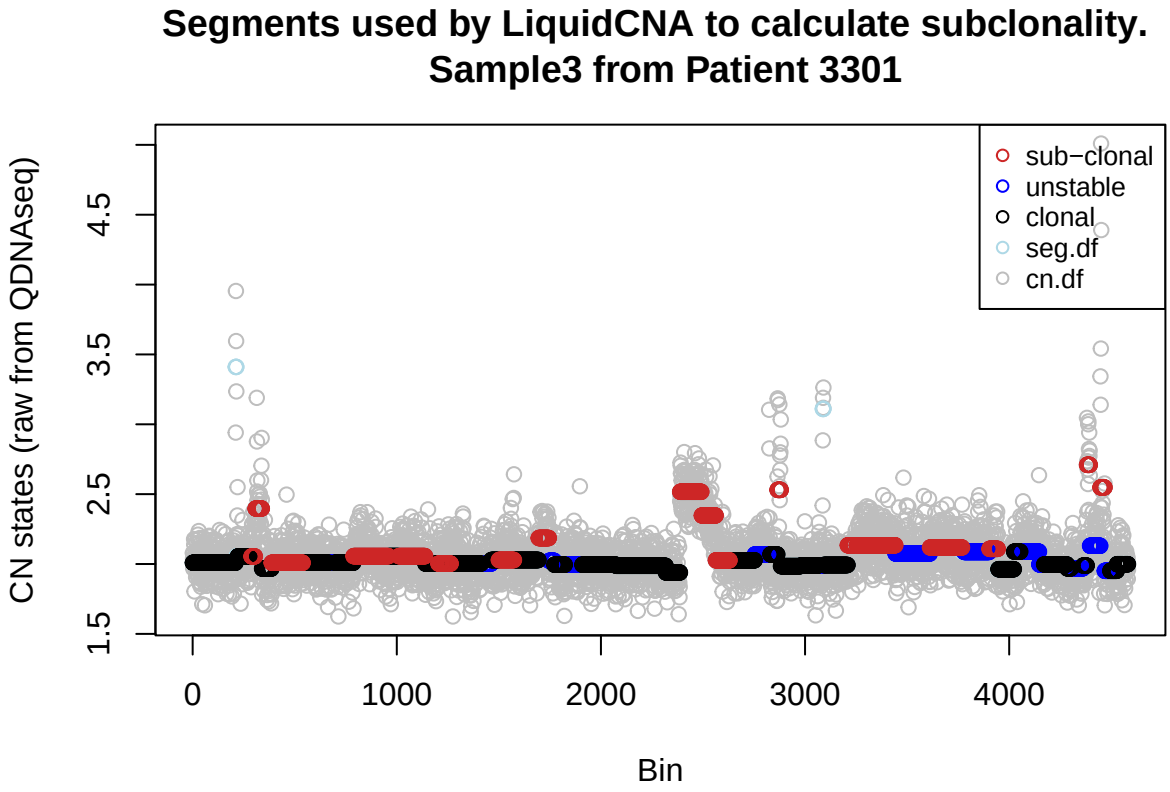
```
## [1] 0.175 0.440 0.195
```

Again, the dCN values for the erroneous sample are at a greater scale.

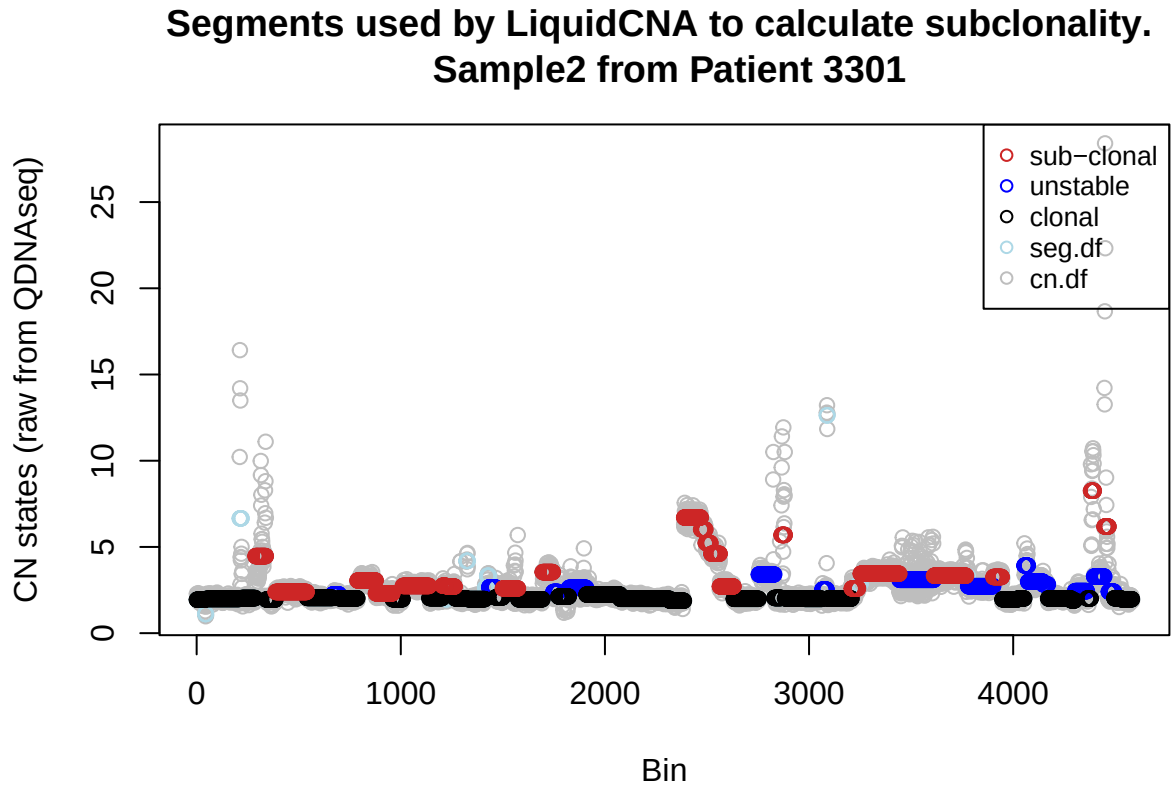
```
head(seg.dcn.toUse.58)
```

```
##       Sample3   Sample2
## 4  0.14544862 2.4674319
## 5  0.35196059 6.2056043
## 7  0.02776333 0.8135755
## 11 0.14727313 1.9618118
## 12 0.07713489 0.6785464
## 14 0.03324136 1.3835290
```

```
plot.3C(3, cn.df.58, seg.df.58, seg.sub.58, seg.plot.58, 3301)
```

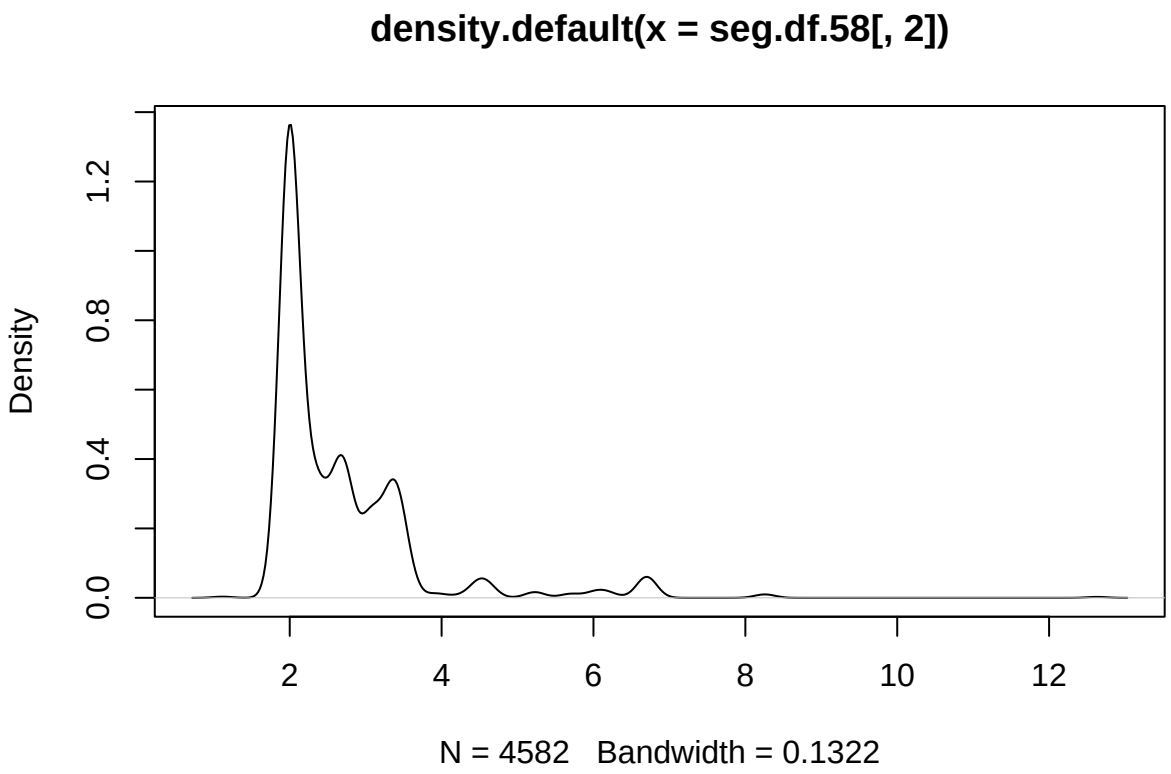



```
plot.3C(2, cn.df.58, seg.df.58, seg.sub.58, seg.plot.58, 3301)
```

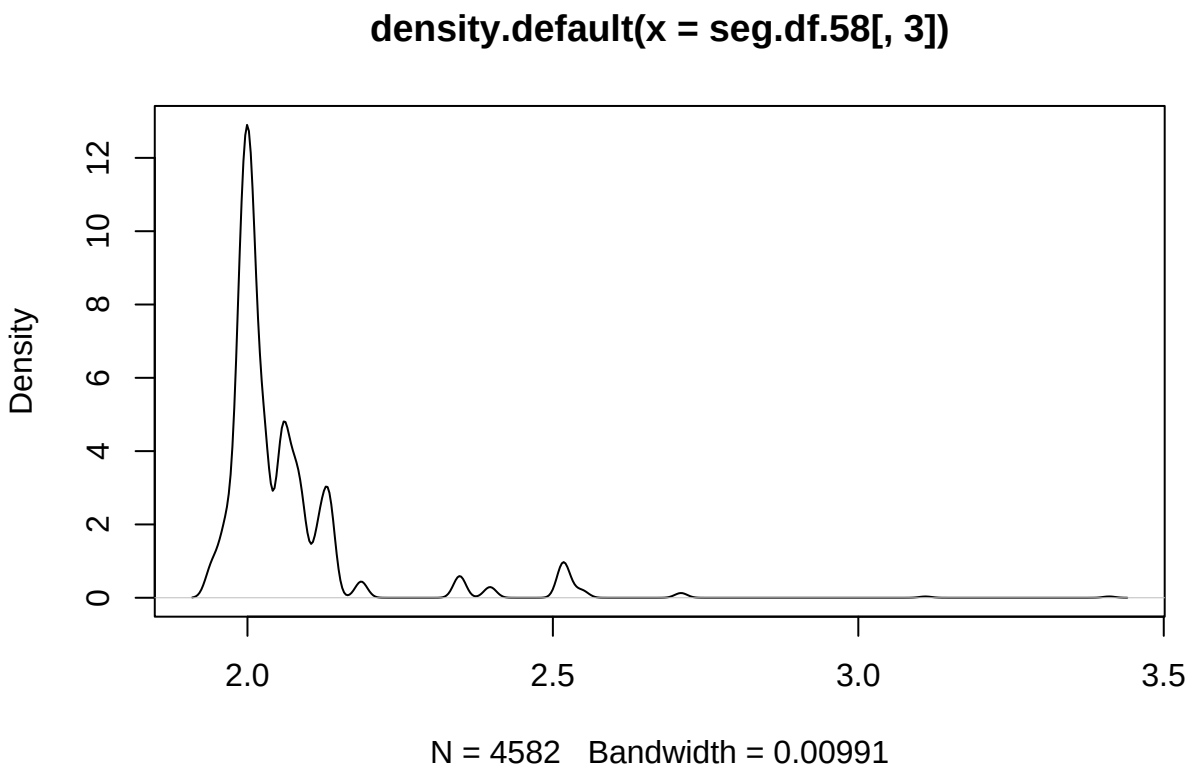


even from the raw input the CN states are highly biased for larger values. Could we trim them out? not consider these segments?

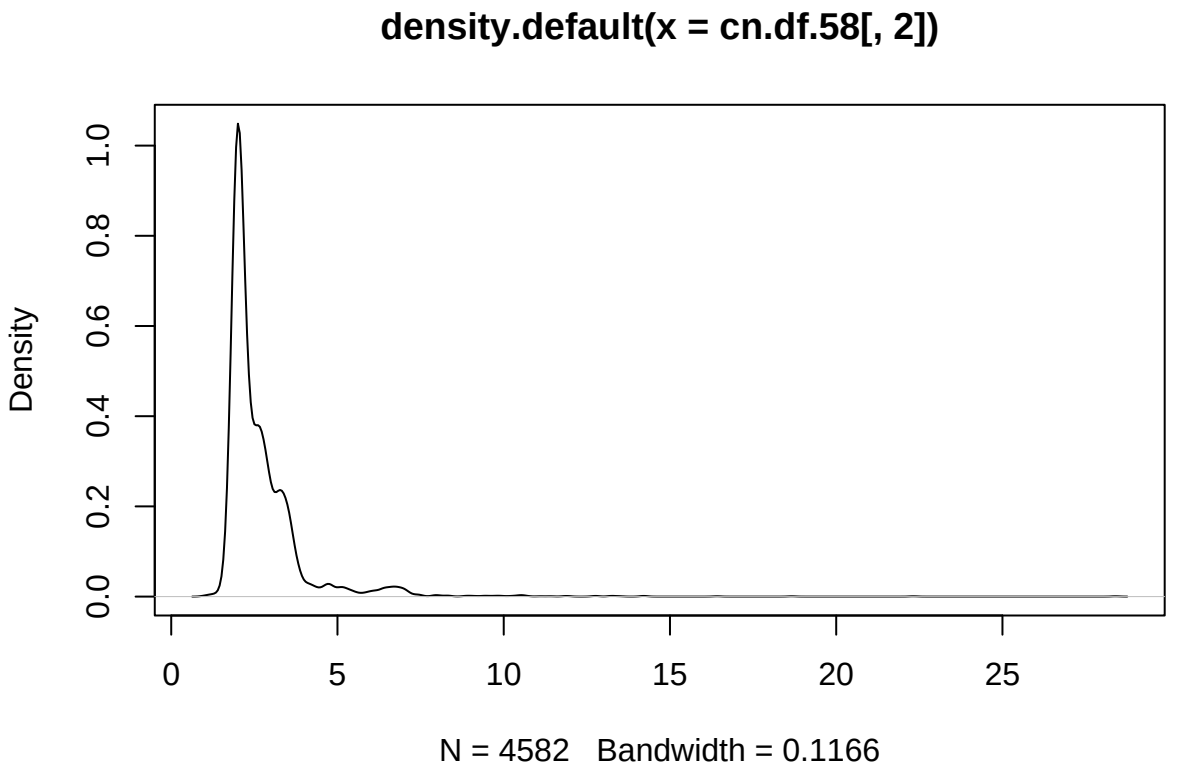
```
plot(density(seg.df.58[,2]))
```



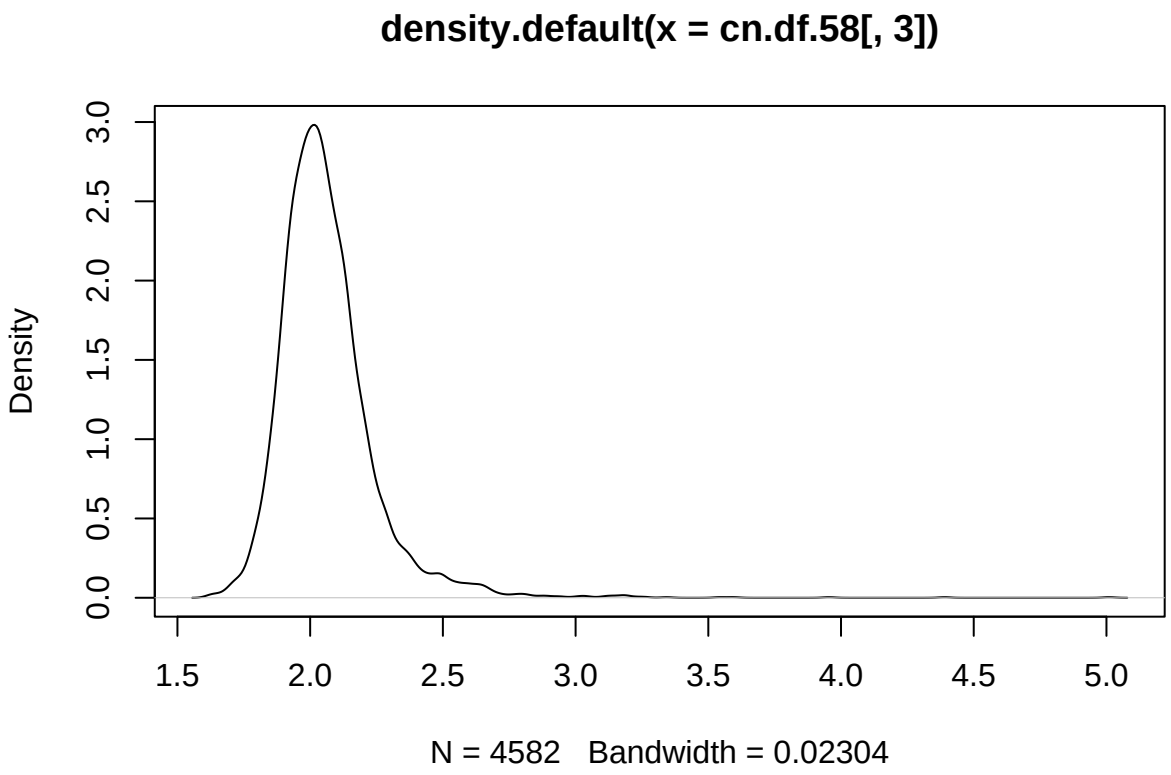
```
plot(density(seg.df.58[,3]))
```




```
plot(density(cn.df.58[,2]))
```



```
plot(density(cn.df.58[,3]))
```

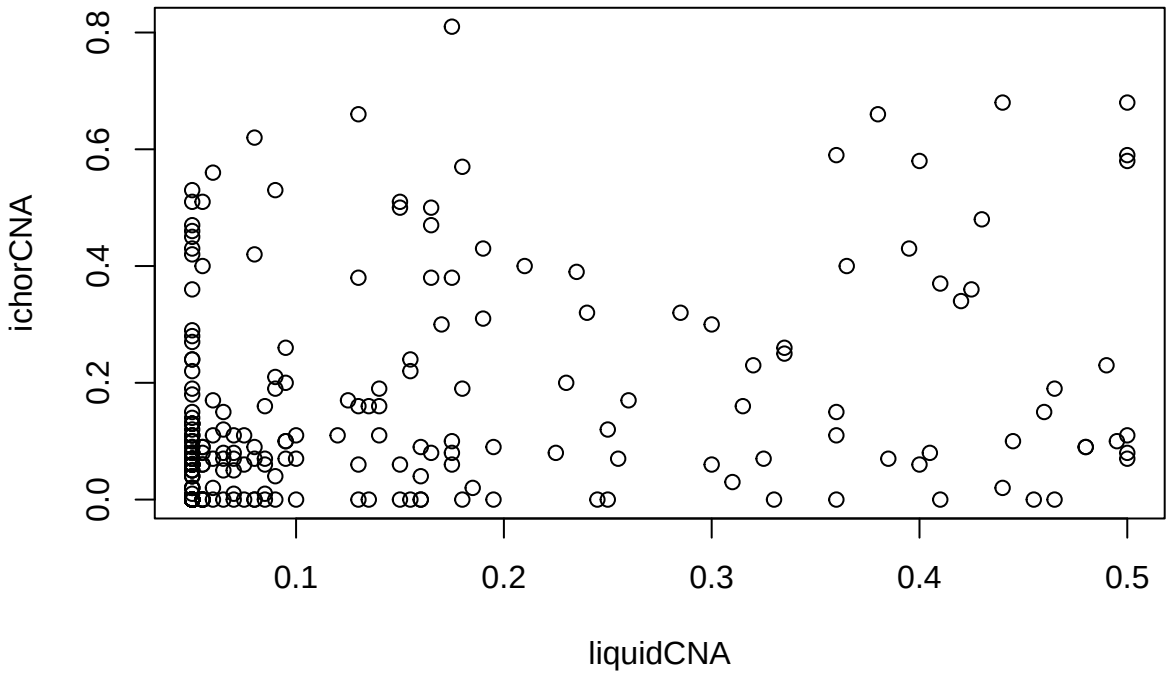


2. ichorCNA

ichorCNA prediction of purity versus liquidCNA

There doesn't seem to be much correlation between the two estimations. Of note, for many of the samples, one of the algorithms estimate purity = 0, whilst the other predict a wide range of purities. For liquidCNA this is expected as baseline samples are restricted/assumed to purity of 0.

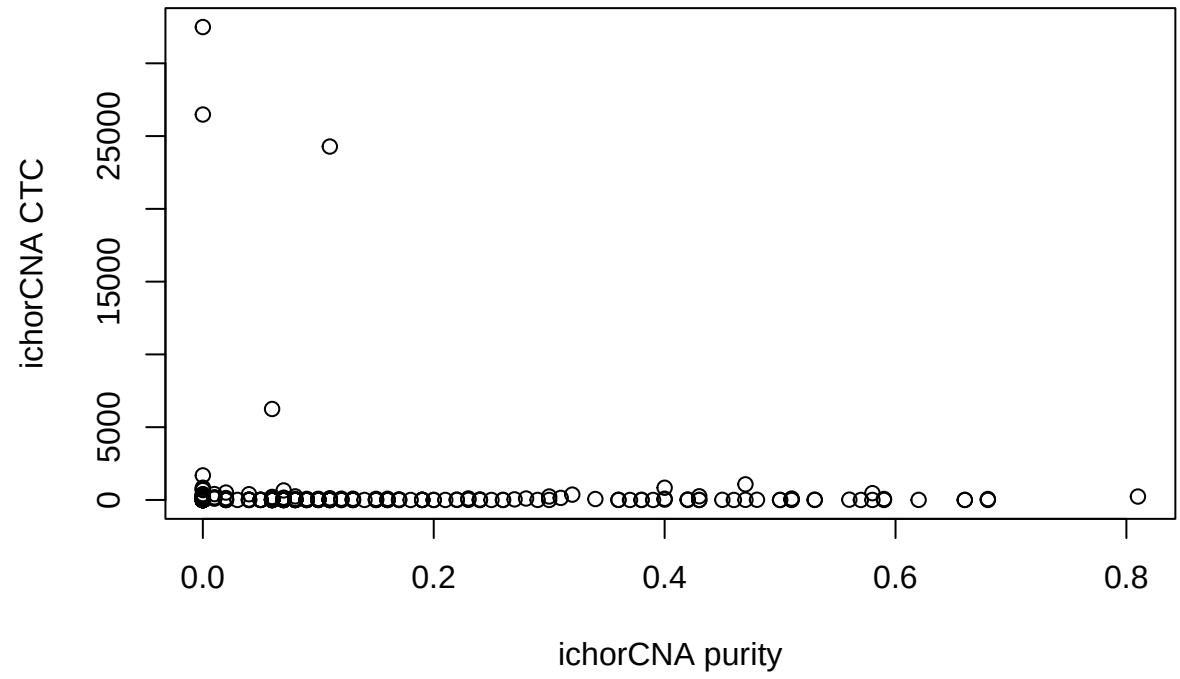
```
optimichor <- ichorCNA$OptimichorCNA
ctc_count <- ichorCNA$CTC_count

ichorCNA.sorted <- ichorCNA[order( ichorCNA$Patient_ID, ichorCNA$Date ),]

ichor.purities <- ichorCNA.sorted$OptimichorCNA
ichor.purities <- ichor.purities / 100

plot(purities, ichor.purities, xlab = "liquidCNA", ylab = "ichorCNA")
```



###

Circulating tumor cell (CTC) count

```
plot(ichor.purities, ctc_count, xlab = "ichorCNA purity", ylab = "ichorCNA CTC")
```



5. Purity