Supplemental Material

Supplementary Figure Legends

Supplementary Figure 1: Source of variants used in this study for DCM Cases (A) and Population Controls (B).

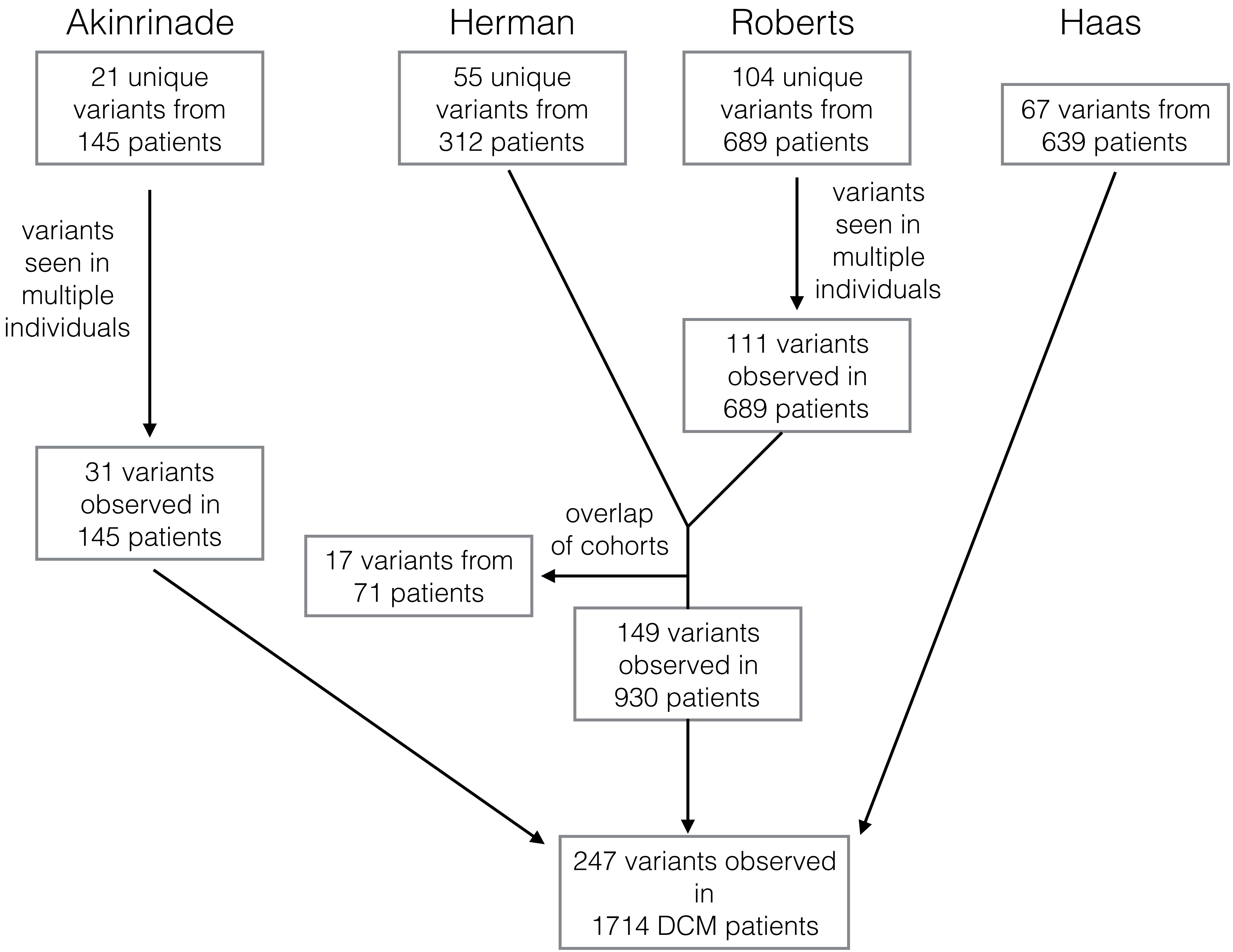
Supplementary Figure 2: TTN truncation mutations in DCM cases are shifted towards higher PSI values. Scatter plots depicting PSI values for exons with truncation mutations seen in DCM and control cohorts. PSI values were estimated from 10 RNA-Seq data sets from human heart tissue. Horizontal jittering was applied to the data to facilitate visualization.

Supplementary Figure 3: The effect of *Cronos* disruption on truncation variant distribution is also seen within the I-band itself. Within constitutive exons (PSI > 0.95) in the I-band, there is also a 4.8-fold increased odds of truncation variants being found in cases vs. controls for those that disrupt the *Cronos* isoform (p=0.006).

Supplementary Figure 4: All previously reported TTN truncations with segregation in families and 31 of 32 mutations in end-stage DCM map to the Group I region, flanked by the Cronos position (dashed line) and the TTN kinase domain. Schematic revealing domain organization of the TTN protein (Ensembl Transcript ID ENST00000589042) as well as the position of TTN truncations demonstrating segregation in families and/or resulting in end-stage DCM.

Supplementary File 1: R Markdown file describing all analyses and including embedded figures.

Supplementary Figure 1A



Akinrinade

21 unique variants from 145 patients

variants seen in multiple individuals

31 variants observed in 145 patients

Herman

55 unique variants from 312 patients

Roberts

104 unique variants from 689 patients

variants seen in multiple individuals

111 variants observed in 689 patients

overlap of cohorts

17 variants from 71 patients

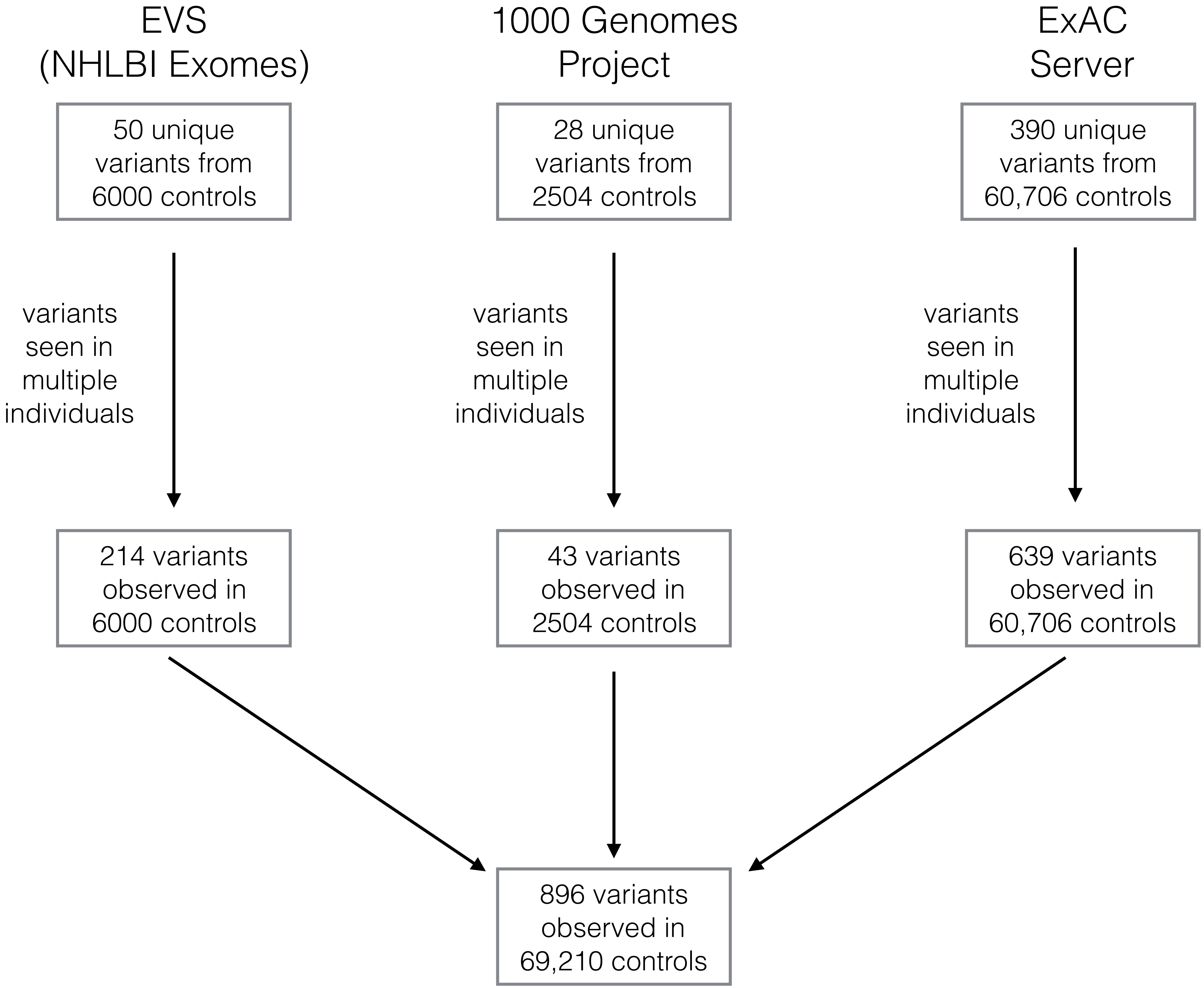
149 variants observed in 930 patients

Haas

67 variants from 639 patients

247 variants observed in 1714 DCM patients

Supplementary Figure 1B



EVS
(NHLBI Exomes)

50 unique variants from 6000 controls

variants seen in multiple individuals

214 variants observed in 6000 controls

1000 Genomes
Project

28 unique variants from 2504 controls

variants seen in multiple individuals

43 variants observed in 2504 controls

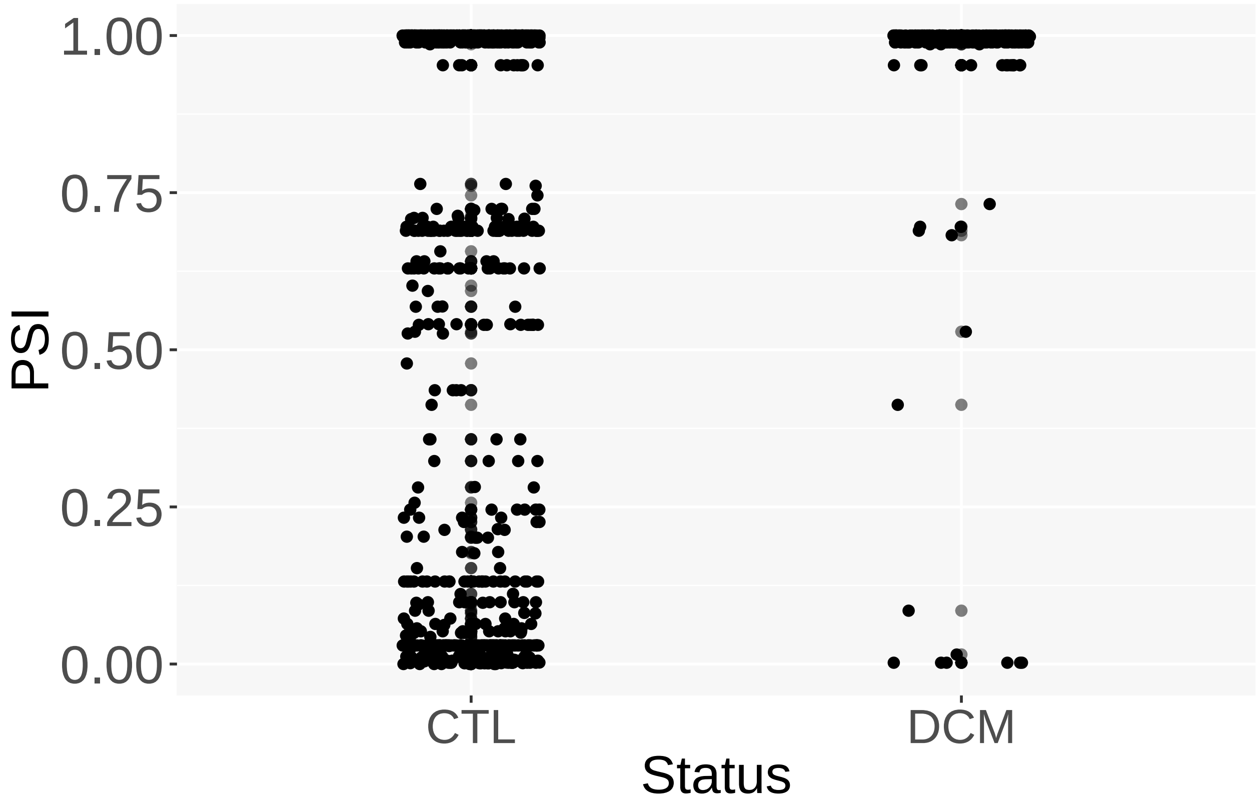
ExAC
Server

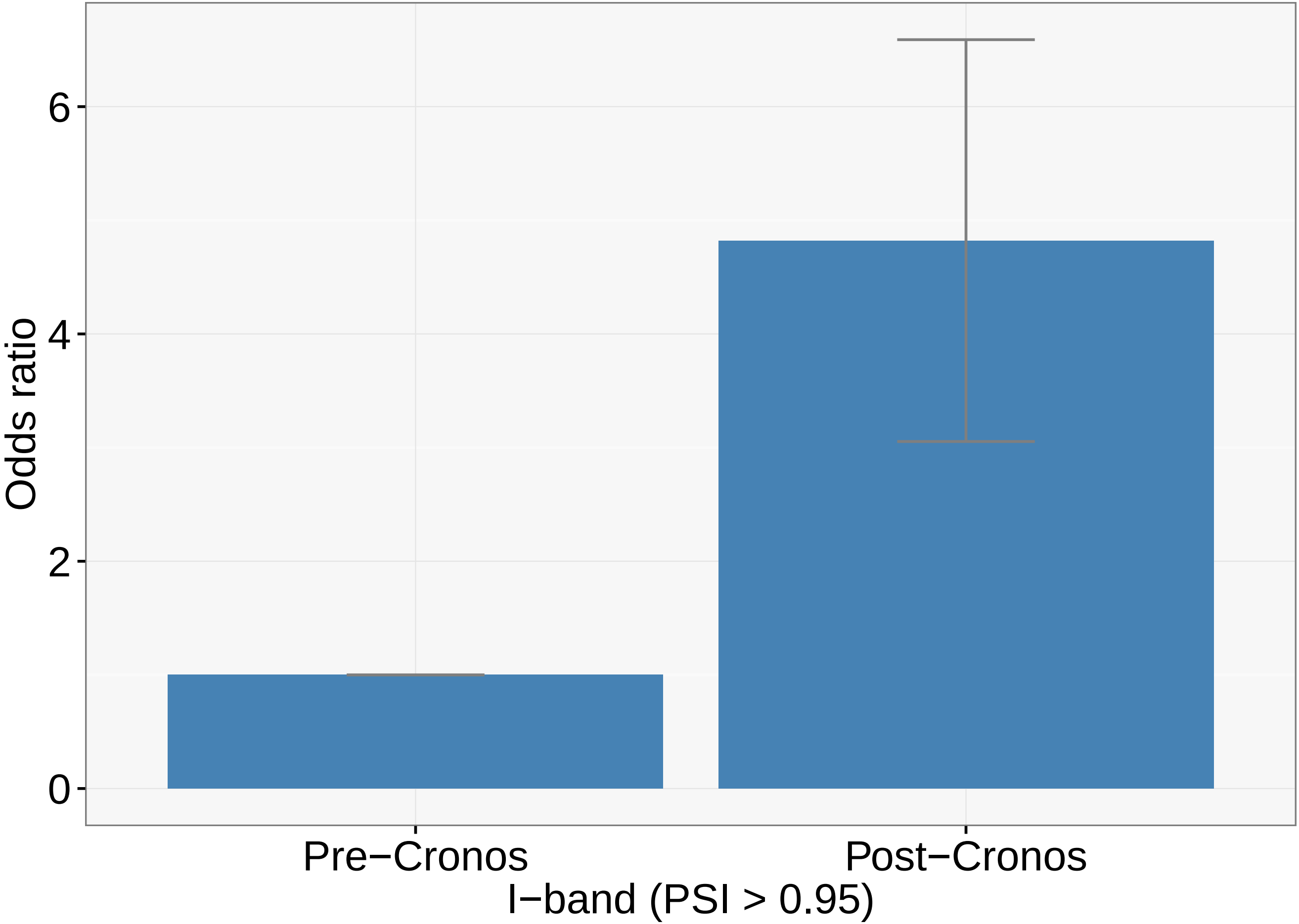
390 unique variants from 60,706 controls

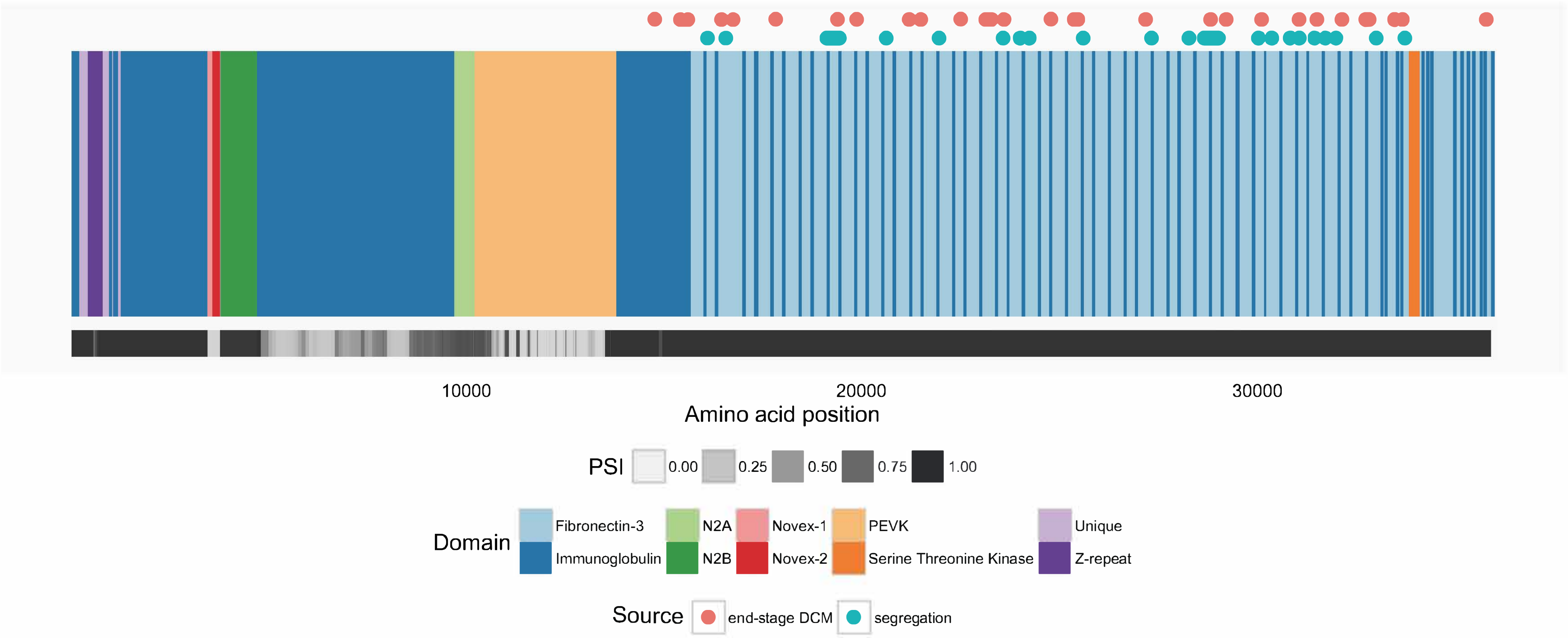
variants seen in multiple individuals

639 variants observed in 60,706 controls

896 variants observed in 69,210 controls


PSI
1.00
0.75
0.50
0.25
0.00
CTL
DCM
Status


Odds ratio
6
4
2
0
Pre−Cronos
Post−Cronos
I−band (PSI > 0.95)


10000
20000
30000
Amino acid position
PSI
0.00
0.25
0.50
0.75
1.00
Domain
Fibronectin-3
Immunoglobulin
N2A
N2B
Novex-1
Novex-2
PEVK
Serine Threonine Kinase
Unique
Z-repeat
Source
end-stage DCM
segregation

TTN Case Control Analysis

The input data consists of lists of TTN truncation variants/mutations of participations in different studies. I do not have individual level data, only the number of individuals in the study. For the control cohorts the number of alleles observed is provided (i.e the number of individuals with the same variant). The same is provided for Akinrinade et al. Roberts et al and Herman et al list the mutation data alongside the participant ID for every person with a TTN truncation variant. Haas et al only lists the variants, but no allele information.

Some description of individual cohorts is provided:

1. Akinrinade: 145 unrelated DCM patients of Finnish origin
2. Haas: 639 patients with sporadic or proven familial DCM enrolled in 8 different clinical centers; unknown if they are unrelated; mutations are listed but number of alleles observed is not.
3. Roberts et al: 374 unrelated idiopathic DCM cases from RBHT hospital; 155 randomly selected end-stage DCM; 163 referred to familial DCM program (unclear if any are related by chance).
4. Herman et al: 92 individuals with DCM from BWH genetics clinic; 71 individuals from UK clinics; 149 individuals with DCM recruited from Italy or Colorado; no explicit mention of related individuals
5. EVS: 6000 individuals, most likely unrelated
6. EXaC: 60,706 individuals, all unrelated
7. 1000 Genomes: 2504 individuals, all unrelated

```
rm(list=ls())
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

```
setwd("/Users/rahuldeo/Dropbox/TTNspl/analysis")
akinrinade <- read.delim("akinrinade_TTN_clean_formatted_psiclean.txt", stringsAsFactors = FALSE)
G1000 <- read.delim("1000G_TTN_snpEff_deleterious_formatted_psiclean.txt", stringsAsFactors = FALSE)
ExAC <- read.delim("ExAC_TTN_deleterious_formatted_psiclean.txt", stringsAsFactors = FALSE)
EVS <- read.delim("EVS_variant_download_GeneName_TTN_deleterious_formatted_psiclean.txt", stringsAsFacto
haas <- read.delim("haas_supplement_refseq_all_newcdna_clean_formatted_psiclean.txt", stringsAsFactors =
herman.ptc <- read.delim("herman_table6_raw_manual_newcdna_clean_formatted_psiclean.txt", stringsAsFacto
herman.spl<- read.delim("herman_table7_raw_manual_newcdna_clean_formatted_psiclean.txt", stringsAsFactor
roberts.rep <- read.delim("roberts_replication_raw_newcdna_formatted_psiclean.txt", stringsAsFactors =
roberts.disc <- read.delim("roberts_ukdiscovery_raw_newcdna_formatted_psiclean.txt", stringsAsFactors =
roberts.endstage <- read.delim("roberts_endstagedcm_raw_newcdna_formatted_psiclean.txt", stringsAsFactor

#look for individuals with shared mutations; there was no active recruitment of families for this study

table(herman.ptc$cDNA_IC)[table(herman.ptc$cDNA_IC)>1]
```

```
## named integer(0)
```

```
table(herman.spl$cDNA_IC)[table(herman.spl$cDNA_IC)>1]
```

```
## named integer(0)
```

```r
table(roberts.rep$cDNA_IC)[table(roberts.rep$cDNA_IC)>1]
```

```
##
## c.76115dupA  c.78991C>T
##           2           2
```

```r
table(roberts.disc$cDNA_IC)[table(roberts.disc$cDNA_IC)>1]
```

```
##
##               c.50170C>T  c.55525_55531delGACAGGA c.81262_81269delCAGATGCT
##                        3                        2                        2
```

```r
table(roberts.endstage$cDNA_IC)[table(roberts.endstage$cDNA_IC)>1]
```

```
## c.100445C>A
##           2
```

```r
roberts.rep[(roberts.rep$cDNA_IC) %in% names(table(roberts.rep$cDNA_IC)[table(roberts.rep$cDNA_IC)>1]),]
```

```
##    CHR  POSITION annotation exon_IC     cDNA_IC      prot_IC AA_IC
## 20   2 179434743 Frameshift      NA c.76115dupA p.Asn25372fs 25372
## 21   2 179434743 Frameshift      NA c.76115dupA p.Asn25372fs 25372
## 23   2 179431868   Nonsense      NA  c.78991C>T  p.Arg26331* 26331
## 24   2 179431868   Nonsense      NA  c.78991C>T  p.Arg26331* 26331
##    exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map       psi
## 20         NA         NA         NA       NA       1  25372 0.9890924
## 21         NA         NA         NA       NA       1  25372 0.9890924
## 23         NA         NA         NA       NA       1  26331 0.9890924
## 24         NA         NA         NA       NA       1  26331 0.9890924
##    domain
## 20 A-band
## 21 A-band
## 23 A-band
## 24 A-band
```

```r
roberts.disc[(roberts.disc$cDNA_IC) %in% names(table(roberts.disc$cDNA_IC)[table(roberts.disc$cDNA_IC)>
```

```
##    CHR  POSITION annotation exon_IC                  cDNA_IC
## 8    2 179429597 Frameshift      NA c.81262_81269delCAGATGCT
## 9    2 179429597 Frameshift      NA c.81262_81269delCAGATGCT
## 41   2 179477082   Nonsense      NA               c.50170C>T
## 42   2 179477082   Nonsense      NA               c.50170C>T
## 43   2 179477082   Nonsense      NA               c.50170C>T
## 48   2 179466199 Frameshift      NA  c.55525_55531delGACAGGA
## 49   2 179466199 Frameshift      NA  c.55525_55531delGACAGGA
##              prot_IC AA_IC exon_Novex cDNA_Novex prot_Novex AA_Novex
## 8   p.Gln27088CysfsX5 27088         NA         NA         NA       NA
## 9   p.Gln27088CysfsX5 27088         NA         NA         NA       NA
## 41        p.Arg16724X 16724         NA         NA         NA       NA
## 42        p.Arg16724X 16724         NA         NA         NA       NA
## 43        p.Arg16724X 16724         NA          NA          NA        NA
## 48 p.Asp18509SerfsX29 18509         NA          NA          NA        NA
## 49 p.Asp18509SerfsX29 18509         NA          NA          NA        NA
##    alleles aa_map       psi domain
## 8        1  27088 0.9890924 A-band
## 9        1  27088 0.9890924 A-band
## 41       1  16724 1.0000000 A-band
## 42       1  16724 1.0000000 A-band
## 43       1  16724 1.0000000 A-band
## 48       1  18509 1.0000000 A-band
## 49       1  18509 1.0000000 A-band
```

```
roberts.endstage[(roberts.endstage$cDNA_IC) %in% names(table(roberts.endstage$cDNA_IC)[table(roberts.en
```

```
##    CHR  POSITION annotation exon_IC     cDNA_IC     prot_IC AA_IC
## 28   2 179401029   Nonsense      NA c.100445C>A   p.S33482* 33482
## 29   2 179401029   Nonsense      NA c.100445C>A p.Ser33482* 33482
##    exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map       psi
## 28         NA         NA         NA       NA       1  33482 0.9927025
## 29         NA         NA         NA       NA       1  33482 0.9927025
##    domain
## 28 A-band
## 29 A-band
```

```
#look for missing values for PSI

G1000[is.na(G1000$psi),]
```

```
##  [1] CHR        POSITION   snpEff     exon_IC    cDNA_IC    prot_IC
##  [7] AA_IC      exon_Novex cDNA_Novex prot_Novex AA_Novex   alleles
## [13] aa_map     psi        domain
## <0 rows> (or 0-length row.names)
```

```
EVS[is.na(EVS$psi),]
```

```
##  [1] CHR        POSITION   annotation exon_IC    cDNA_IC    prot_IC
##  [7] AA_IC      exon_Novex cDNA_Novex prot_Novex AA_Novex   alleles
## [13] aa_map     psi        domain
## <0 rows> (or 0-length row.names)
```

```
ExAC[is.na(ExAC$psi),]
```

```
##     CHR  POSITION   annotation exon_IC                     cDNA_IC prot_IC
## 11    2 179394966 splice donor      NA              c.106374+1delG
## 212   2 179532167 splice donor      NA  c.35713+1_35713+2delGTinsT
## 213   2 179532167 splice donor      NA c.35713+1_35713+2delGTinsGC
##     AA_IC exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map psi
## 11  35458         NA         NA         NA       NA       1  35458  NA
## 212 11904         NA         NA         NA       NA       4  11904  NA
## 213 11904         NA         NA         NA       NA       1  11904  NA
##     domain
## 11  M-line
## 212 I-band
## 213 I-band
```

```
haas[is.na(haas$psi),]
```

```
##  [1] CHR        POSITION   annotation exon_IC    cDNA_IC    prot_IC   
##  [7] AA_IC      exon_Novex cDNA_Novex prot_Novex AA_Novex   alleles   
## [13] aa_map     psi        domain    
## <0 rows> (or 0-length row.names)
```

```
akinrinade[is.na(akinrinade$psi),]
```

```
##    CHR  POSITION annotation exon_IC               cDNA_IC         prot_IC
## 11   2 179447666 frameshift      NA c.65860_65863dupTTAG D21955VfsX21957
##    AA_IC exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map psi
## 11 21955         NA         NA         NA       NA       1  21955  NA
##    domain
## 11 A-band
```

```
herman.spl[is.na(herman.spl$psi),]
```

```
##  [1] CHR        POSITION   annotation exon_IC    cDNA_IC    prot_IC   
##  [7] AA_IC      exon_Novex cDNA_Novex prot_Novex AA_Novex   alleles   
## [13] aa_map     psi        domain    
## <0 rows> (or 0-length row.names)
```

```
herman.ptc[is.na(herman.ptc$psi),]
```

```
##  [1] CHR        POSITION   annotation exon_IC    cDNA_IC    prot_IC   
##  [7] AA_IC      exon_Novex cDNA_Novex prot_Novex AA_Novex   alleles   
## [13] aa_map     psi        domain    
## <0 rows> (or 0-length row.names)
```

```
roberts.rep[is.na(roberts.rep$psi),]
```

```
##  [1] CHR        POSITION   annotation exon_IC    cDNA_IC    prot_IC   
##  [7] AA_IC      exon_Novex cDNA_Novex prot_Novex AA_Novex   alleles   
## [13] aa_map     psi        domain    
## <0 rows> (or 0-length row.names)
```

```
roberts.disc[is.na(roberts.disc$psi),]
```

```
##  [1] CHR        POSITION   annotation exon_IC    cDNA_IC    prot_IC   
##  [7] AA_IC      exon_Novex cDNA_Novex prot_Novex AA_Novex   alleles   
## [13] aa_map     psi        domain    
## <0 rows> (or 0-length row.names)
```

```
roberts.endstage[is.na(roberts.endstage$psi),]
```

```
##  [1] CHR        POSITION   annotation exon_IC    cDNA_IC    prot_IC   
##  [7] AA_IC      exon_Novex cDNA_Novex prot_Novex AA_Novex   alleles   
## [13] aa_map     psi        domain    
## <0 rows> (or 0-length row.names)
```

```
#correct by manual look-up
ExAC$psi[11] = 0.998
ExAC$psi[212] = 0.0012
ExAC$psi[213] = 0.0012
akinrinade$psi[11] = 1

#correct error of amino acid assignment
ExAC$aa_map[369] = 1044
```

Thus 7 patients in the Roberts manuscript appeared to share the same mutation as others in the same sub-cohort. However, these may still be unrelated - and we will leave it as such.

Look for cross-duplicates b/w Herman and Roberts as Cohort B from Herman et al appears to overlap with the Roberts

```
herman.all <- rbind.data.frame(herman.spl, herman.ptc)
roberts.all <- rbind.data.frame(roberts.rep, roberts.disc, roberts.endstage)

herman.all[herman.all$cDNA_IC %in% roberts.all$cDNA_IC,]
```

```
##    CHR  POSITION      annotation exon_IC      cDNA_IC    prot_IC AA_IC
## 6    2 179457005 splice-acceptor      NA c.59627-1G>A p.Asp19875 19875
## 8    2 179441649    splice-donor      NA c.69412+1G>A p.Gly23137 23137
## 13   2 179401029     stop gained      NA  c.100445C>A    S33482X 33482
## 15   2 179404286     stop gained      NA   c.98506C>T    R32836X 32836
## 22   2 179413187     stop gained      NA   c.93166C>T    R31056X 31056
## 25   2 179422457     stop gained      NA   c.87624C>A    Y29208X 29208
## 37   2 179444429     stop gained      NA   c.67495C>T    R22499X 22499
## 38   2 179452435     stop gained      NA   c.63601C>T    R21201X 21201
## 40   2 179454957     stop gained      NA   c.61495C>T    R20499X 20499
## 48   2 179471841     stop gained      NA   c.53488G>T    G17830X 17830
## 51   2 179485012     stop gained      NA   c.46236C>A    C15412X 15412
##    exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map       psi
## 6          NA         NA         NA       NA       1  19875 1.0000000
## 8          NA         NA         NA       NA       1  23137 1.0000000
## 13         NA         NA         NA       NA       1  33482 0.9927025
## 15         NA         NA         NA       NA       1  32836 0.9993624
## 22         NA         NA         NA       NA       1  31056 0.9962754
## 25         NA         NA         NA       NA       1  29208 0.9915163
## 37         NA         NA         NA       NA       1  22499 1.0000000
## 38         NA         NA         NA       NA       1  21201 1.0000000
## 40         NA         NA         NA       NA       1  20499 1.0000000
## 48         NA         NA         NA       NA       1  17830 1.0000000
## 51         NA         NA         NA       NA       1  15412 1.0000000
##    domain
## 6  A-band
## 8  A-band
## 13 A-band
## 15 A-band
## 22 A-band
## 25 A-band
## 37 A-band
## 38 A-band
## 40 A-band
## 48 A-band
## 51 I-band
```

```
#confirmed that these have UK identifiers, number 40 does not

#check select overlap from manual examination of Herman et al supplement
roberts.all[roberts.all$POSITION == 179408239,]
```

```
##     CHR  POSITION annotation exon_IC           cDNA_IC    prot_IC AA_IC
## 104   2 179408239 Frameshift      NA c.96460_96461insA p.T32154fs 32154
##     exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map       psi
## 104         NA         NA         NA       NA       1  32154 0.9988934
##     domain
## 104 A-band
```

```
roberts.all[roberts.all$POSITION == 179417723,]
```

```
##  [1] CHR        POSITION   annotation exon_IC    cDNA_IC    prot_IC   
##  [7] AA_IC      exon_Novex cDNA_Novex prot_Novex AA_Novex   alleles   
## [13] aa_map     psi        domain    
## <0 rows> (or 0-length row.names)
```

```
roberts.all[roberts.all$POSITION == 179424398,]
```

```
##    CHR  POSITION annotation exon_IC            cDNA_IC    prot_IC AA_IC
## 99   2 179424398 Frameshift      NA c.86459_86460delCT p.S28820fs 28820
##    exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map       psi
## 99         NA         NA         NA       NA       1  28820 0.9890924
##    domain
## 99 A-band
```

```
roberts.all[roberts.all$POSITION == 179440067,]
```

```
##    CHR  POSITION annotation exon_IC           cDNA_IC    prot_IC AA_IC
## 94   2 179440067 Frameshift      NA c.70791_70791delA p.E23597fs 23597
##    exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map       psi
## 94         NA         NA         NA       NA       1  23597 0.9890924
##    domain
## 94 A-band
```

```
roberts.all[roberts.all$POSITION == 179441015,]
```

```
##    CHR  POSITION annotation exon_IC           cDNA_IC    prot_IC AA_IC
## 93   2 179441015 Frameshift      NA c.69843_69843delA p.K23281fs 23281
##    exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map       psi
## 93         NA         NA         NA       NA       1  23281 0.9890924
##    domain
## 93 A-band
```

```
roberts.all[roberts.all$POSITION == 179477004,]
```

```
##    CHR  POSITION annotation exon_IC           cDNA_IC    prot_IC AA_IC
## 85   2 179477004 Frameshift      NA c.50247_50247delT p.F16749fs 16749
##    exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map psi domain
## 85         NA         NA         NA       NA       1  16749   1 A-band
```

```
roberts.all[roberts.all$POSITION == 179401029,]
```

```
##     CHR  POSITION annotation exon_IC     cDNA_IC     prot_IC AA_IC
## 108   2 179401029   Nonsense      NA c.100445C>A   p.S33482* 33482
## 109   2 179401029   Nonsense      NA c.100445C>A p.Ser33482* 33482
##     exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map       psi
## 108         NA         NA         NA       NA       1  33482 0.9927025
## 109         NA         NA         NA       NA       1  33482 0.9927025
##     domain
## 108 A-band
## 109 A-band
```

```
herman.all <- herman.all[-c(6,8,13,15,22,25,37,38,48,51),]
roberts.all <- roberts.all[-c(104,99,94,93,85,108,109),]
```

Combine DCM and CTL data sets. Expand the CTL and DCM data sets since there are multiple individuals with the same variants.

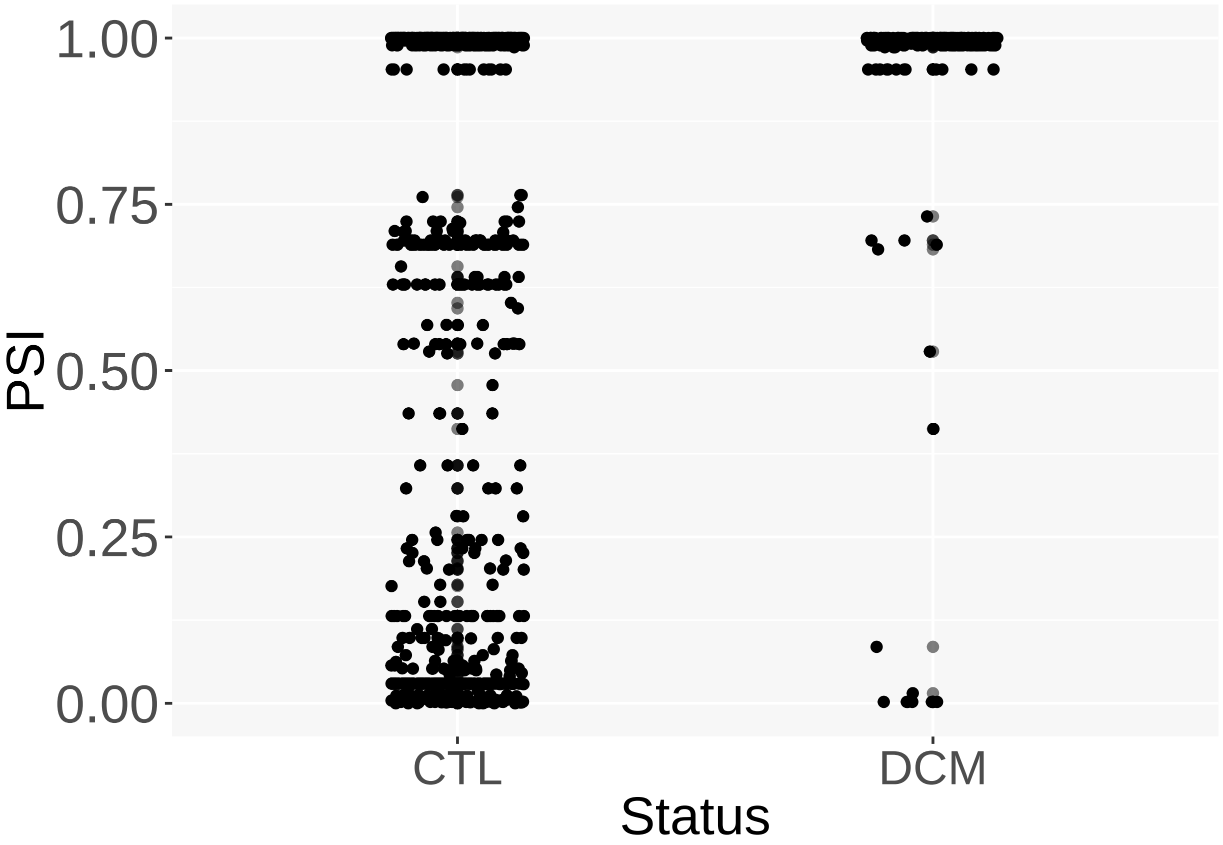
```
##   CHR  POSITION  annotation exon_IC            cDNA_IC         prot_IC
## 1   2 179419765 stop gained      NA         c.88421G>A         W29474X
## 2   2 179423146 stop gained      NA         c.87040C>T         R29014X
## 3   2 179430320 stop gained      NA         c.80539C>T         Q26847X
## 4   2 179431415  frameshift      NA        c.79443delC C26482VfsX26497
## 5   2 179433665 stop gained      NA         c.77194C>T         Q25732X
## 6   2 179434009  frameshift      NA c.76849_76850insGT S25617VfsX25634
##   AA_IC exon_Novex cDNA_Novex prot_Novex AA_Novex alleles aa_map       psi
## 1 29474         NA         NA         NA       NA       3  29474 0.9861176
## 2 29014         NA         NA         NA       NA       1  29014 0.9898195
## 3 26847         NA         NA         NA       NA       1  26847 0.9890924
## 4 26482         NA         NA         NA       NA       1  26482 0.9890924
## 5 25732         NA         NA         NA       NA       1  25732 0.9890924
## 6 25617         NA         NA         NA       NA       2  25617 0.9890924
##   domain status
## 1 A-band    DCM
## 2 A-band    DCM
## 3 A-band    DCM
## 4 A-band    DCM
## 5 A-band    DCM
## 6 A-band    DCM
```

```
##   CHR  POSITION                             annotation exon_IC
## 1   2 179393000    splice_donor_variant&intron_variant 361/362
## 2   2 179393524                            stop_gained 360/363
## 3   2 179400577 splice_acceptor_variant&intron_variant 357/362
## 4   2 179404241                            stop_gained 352/363
## 5   2 179411199                            stop_gained 342/363
## 6   2 179412199                            stop_gained 339/363
##         cDNA_IC     prot_IC       AA_IC exon_Novex cDNA_Novex prot_Novex
## 1 c.107377+1G>A                   35792         NA         NA         NA
## 2   c.106954C>T p.Arg35652* 35652/35991         NA         NA         NA
## 3 c.100766-1G>T                   33588         NA         NA         NA
## 4    c.98551C>T p.Arg32851* 32851/35991         NA         NA         NA
## 5    c.94859T>G p.Leu31620* 31620/35991         NA         NA         NA
## 6    c.94154C>G p.Ser31385* 31385/35991         NA         NA         NA
##   AA_Novex alleles aa_map       psi domain status
## 1       NA       1  35792 0.9999133 M-line    CTL
## 2       NA       1  35652 0.9988036 M-line    CTL
## 3       NA       1  33588 0.9983953 A-band    CTL
## 4       NA       1  32851 0.9993624 A-band    CTL
## 5       NA       1  31620 0.9985317 A-band    CTL
## 6       NA       1  31385 0.9962754 A-band    CTL
```

We are trying to understand what characteristics of mutations differentiate those found in cases vs. controls. The response variable is whether the mutation is found in a case or a control. The predictors are characteristics about each mutation. The number of individuals from which these mutations were derived is not important for this analysis.

Look at the relationship of case and control status with degree of alternative splicing (PSI). We will also generate factors for traditional divisions based on electron micrographs (e.g. A-band), and note the position of the Cronos Isoform

```r
DCM.CTL.all <- rbind.data.frame(CTL.all.rep, DCM.all.rep)

DCM.CTL.all$status <- factor(DCM.CTL.all$status, levels = c("CTL", "DCM"))
DCM.CTL.all$cronos <- DCM.CTL.all$aa_map > 14761
DCM.CTL.all$const <- DCM.CTL.all$psi > 0.95
DCM.CTL.all$domain <- factor(DCM.CTL.all$domain, levels =c("Z-disk","I-band","A-band", "M-line"))

p <- ggplot(DCM.CTL.all, aes(x = status, y = psi))
p <- p + geom_point(alpha = 0.5) + geom_jitter(width = 0.35)
p<-p +theme(axis.title.x = element_text(size = 20), axis.title.y = element_text(size = 20), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p <- p + xlab("Status") + ylab("PSI")
p
```



```
ggsave("psi_variation_DCM_CTL.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

Estimate odds ratios for individual PSI bins

```
DCM.CTL.all$psiexp <- rep(NA)
for (i in 1:nrow(DCM.CTL.all))
{
  if (DCM.CTL.all$psi[i] < 0.4) {
    DCM.CTL.all$psiexp[i] = 0
   } else if (DCM.CTL.all$psi[i] <0.65) {
    DCM.CTL.all$psiexp[i] = 1
   } else if (DCM.CTL.all$psi[i] <0.75) {
    DCM.CTL.all$psiexp[i] = 2
   } else if (DCM.CTL.all$psi[i] <1.01) {
    DCM.CTL.all$psiexp[i] = 3
   } else {
    DCM.CTL.all$psiexp[i] = 4
   }
}

DCM.CTL.all$psiexp <- factor(DCM.CTL.all$psiexp)

#Look at the distribution of individuals in each bin
table(DCM.CTL.all$psiexp)
```

```
##
##   0   1   2   3
## 344  57  80 662
```

```
model.psi <- glm(status ~ psi, family = binomial(link = "logit"), data = DCM.CTL.all)
summary(model.psi)
```

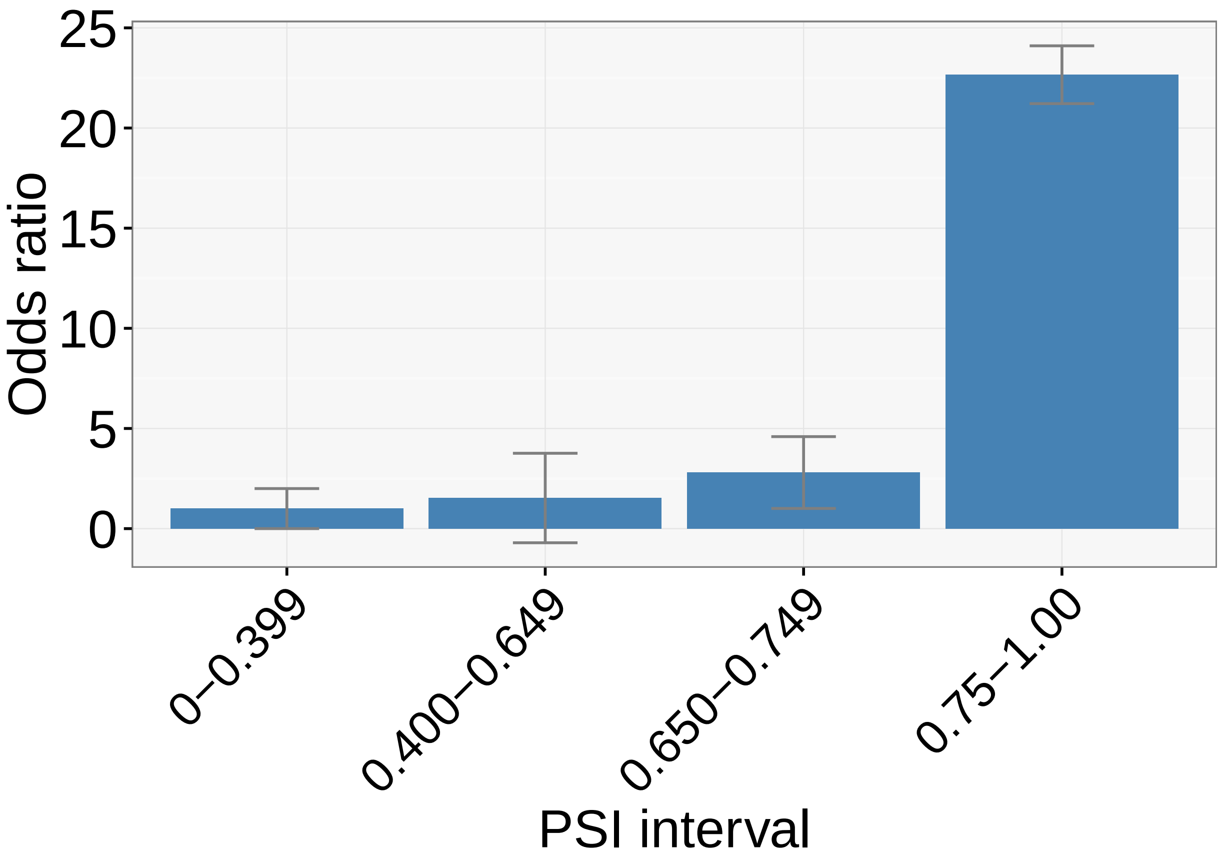
```
##
## Call:
## glm(formula = status ~ psi, family = binomial(link = "logit"),
##     data = DCM.CTL.all)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9171  -0.9154  -0.2523  -0.1580   2.9655
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.3926     0.4000 -10.981   <2e-16 ***
## psi           3.7440     0.4192   8.931   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1193.1  on 1142  degrees of freedom
## Residual deviance: 1005.3  on 1141  degrees of freedom
## AIC: 1009.3
##
## Number of Fisher Scoring iterations: 6
```

```
model.psi.bin <- glm(status ~ psiexp, family = binomial(link = "logit"), data = DCM.CTL.all)
summary(model.psi.bin)
```

```
##
## Call:
## glm(formula = status ~ psiexp, family = binomial(link = "logit"),
##     data = DCM.CTL.all)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9290  -0.9290  -0.2169  -0.2169   2.7427
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.7377     0.3577 -10.448   <2e-16 ***
## psiexp1       0.4235     0.8038   0.527    0.598
## psiexp2       1.0296     0.5842   1.762    0.078 .
## psiexp3       3.1206     0.3669   8.506   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1193.12  on 1142  degrees of freedom
## Residual deviance:  988.32  on 1139  degrees of freedom
## AIC: 996.32
##
## Number of Fisher Scoring iterations: 6
```

```
psi.coeff <- c(0, model.psi.bin$coeff[2:4])
psi.se <- c(0, summary(model.psi.bin)$coeff[,2][2:4])
psinames <- c("0-0.399", "0.400-0.649", "0.650-0.749", "0.75-1.00")
psi.all <- cbind.data.frame(psinames, psi.coeff, psi.se)
limits <- aes(ymax = exp(psi.coeff) + exp(psi.se), ymin=exp(psi.coeff) - exp(psi.se))
dodge <- position_dodge(width=0.9)


p <- ggplot(psi.all, aes(x = psinames, y = exp(psi.coeff)))
p <- p + geom_bar(stat = "identity", fill = "steelblue")  + ylab("Odds ratio") + xlab("PSI interval")
p <- p + geom_errorbar(limits, position = position_dodge(width=0.9), width = 0.25, color = "gray50")
p <- p + theme_bw() + scale_fill_brewer(type = "qual", palette = 1)
p<-p +theme(axis.title.x = element_text(size = 20), axis.title.y = element_text(size = 20), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p
```



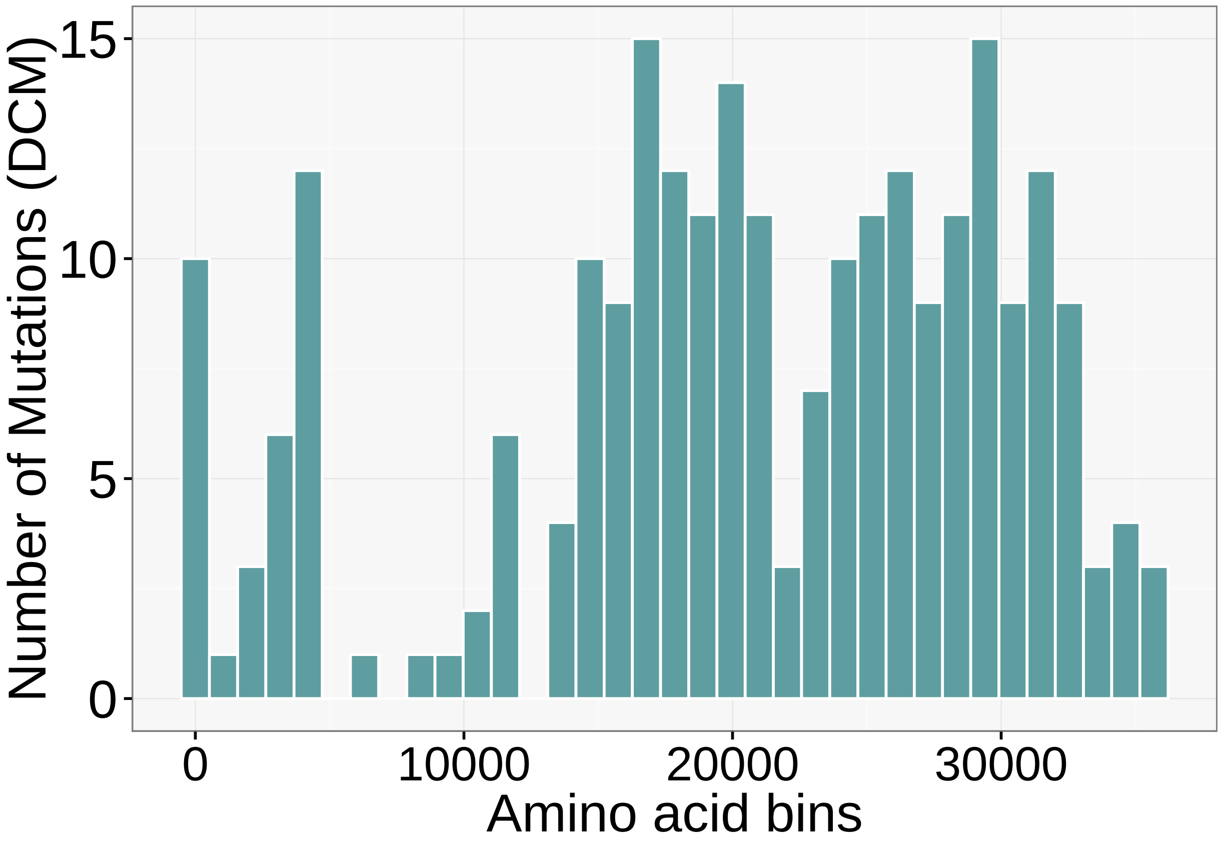
```
ggsave("comparison_of_DCM_vs_CTL_mutation_distribution_psi_bin_nocronosadj.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

Analysis of variation of Case-Control status with mutation position along the length of the protein.

Amino acids are grouped into bins of 2000 amino acids.

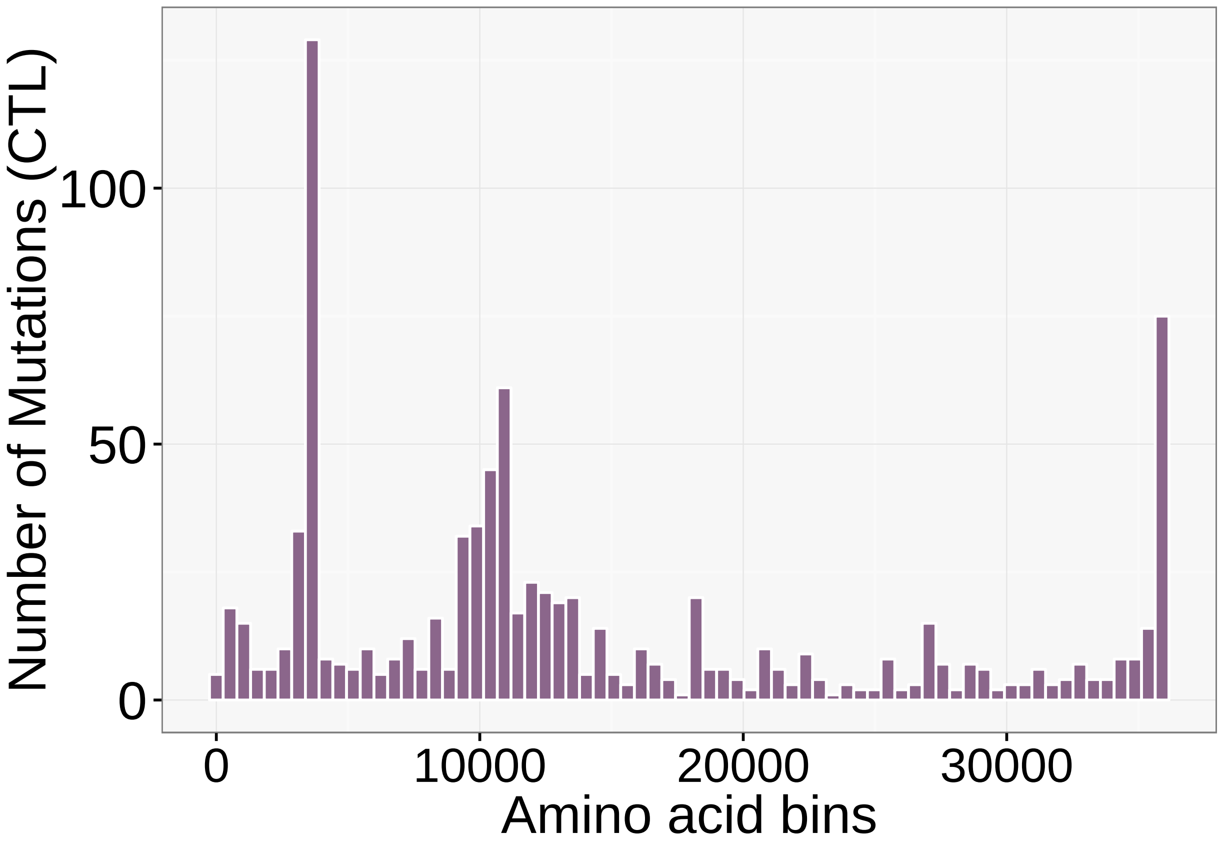
```
DCM.CTL.all$aabin <- rep(NA)

for (i in 1:nrow(DCM.CTL.all))
{
  DCM.CTL.all$aabin[i] = floor(DCM.CTL.all$aa_map[i]/2000)
}

DCM.CTL.all$aabin <- factor(DCM.CTL.all$aabin)

p <- ggplot(DCM.CTL.all[DCM.CTL.all$status=="DCM",], aes(x = aa_map))
p <- p + geom_histogram(fill = "cadet blue", bins=35, colour = "white")  + ylab("Number of Mutations (D
p <- p + theme_bw() + scale_fill_brewer(type = "qual", palette = 1)
p<-p +theme(axis.title.x = element_text(size = 20), axis.title.y = element_text(size = 20), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p
```



```
ggsave("histogram_distribution_mutations_DCM.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
p <- ggplot(DCM.CTL.all[DCM.CTL.all$status=="CTL",], aes(x = aa_map))
p <- p + geom_histogram(fill = "plum4", bins=70, colour = "white")  + ylab("Number of Mutations (CTL)")
p <- p + theme_bw() + scale_fill_brewer(type = "qual", palette = 1)
p<-p +theme(axis.title.x = element_text(size = 20), axis.title.y = element_text(size = 20), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p
```



```
ggsave("histogram_distribution_mutations_CTL.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

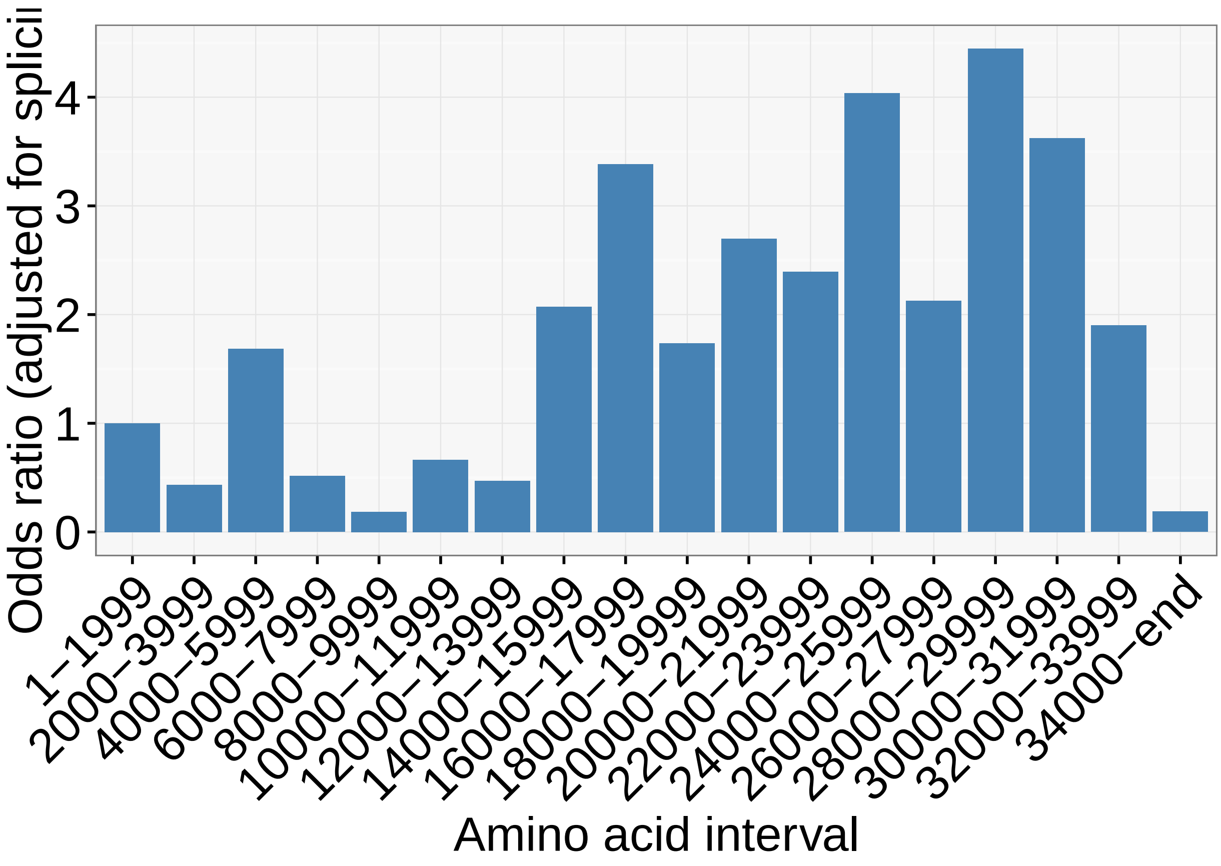
Plot odds ratios for individual bins

```
model.aabin <- glm(status ~ psi + aabin, family = binomial(link = "logit"), data = DCM.CTL.all)
summary(model.aabin)
```

```
##
## Call:
## glm(formula = status ~ psi + aabin, family = binomial(link = "logit"),
##     data = DCM.CTL.all)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3664  -0.5337  -0.2511  -0.1816   3.1097
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  -3.3319     0.5807  -5.738 9.58e-09 ***
## psi           2.2873     0.5181   4.415 1.01e-05 ***
## aabin1       -0.8321     0.4696  -1.772 0.076420 .  
## aabin2        0.5218     0.5168   1.010 0.312657    
## aabin3       -0.6626     1.1219  -0.591 0.554804    
## aabin4       -1.6894     0.8023  -2.106 0.035228 *  
## aabin5       -0.4096     0.5428  -0.755 0.450495    
## aabin6       -0.7559     0.6274  -1.205 0.228298    
## aabin7        0.7287     0.4451   1.637 0.101570    
## aabin8        1.2190     0.4361   2.795 0.005185 ** 
## aabin9        0.5521     0.4192   1.317 0.187861    
## aabin10       0.9933     0.4530   2.193 0.028312 *  
## aabin11       0.8729     0.4739   1.842 0.065516 .  
## aabin12       1.3950     0.4844   2.880 0.003982 ** 
## aabin13       0.7559     0.4397   1.719 0.085561 .  
## aabin14       1.4920     0.4467   3.340 0.000837 ***
## aabin15       1.2877     0.4709   2.735 0.006245 ** 
## aabin16       0.6422     0.4913   1.307 0.191167    
## aabin17      -1.6697     0.5048  -3.308 0.000940 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 1193.12  on 1142  degrees of freedom
## Residual deviance:  879.13  on 1124  degrees of freedom
## AIC: 917.13
## 
## Number of Fisher Scoring iterations: 6
```

```
aa.coeff <- c(0, model.aabin$coeff[3:19])
aa.se <- c(0, summary(model.aabin)$coeff[,2][3:19])
aanames <- c("1-1999",
             "2000-3999",
             "4000-5999",
             "6000-7999",
             "8000-9999",
             "10000-11999",
             "12000-13999",
             "14000-15999",
             "16000-17999",
             "18000-19999",
             "20000-21999",
             "22000-23999",
             "24000-25999",
             "26000-27999",
             "28000-29999",
             "30000-31999",
             "32000-33999",
             "34000-end")

aa.all <- cbind.data.frame(aanames, aa.coeff, aa.se)
aa.all$aanames <- factor(aa.all$aanames, levels = aanames)
limits <- aes(ymax = exp(aa.coeff) + exp(aa.se), ymin=exp(aa.coeff) - exp(aa.se))
dodge <- position_dodge(width=0.9)

p <- ggplot(aa.all, aes(x = aanames, y = exp(aa.coeff)))
p <- p + geom_bar(stat = "identity", fill = "steelblue")  + ylab("Odds ratio (adjusted for splicing)") +
p <- p + theme_bw() + scale_fill_brewer(type = "qual", palette = 1)
p<-p +theme(axis.title.x = element_text(size = 18), axis.title.y = element_text(size = 18), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p
```



```
ggsave("comparison_of_case_vs_control_mutation_distribution_2000aa_position_bin_noSE.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

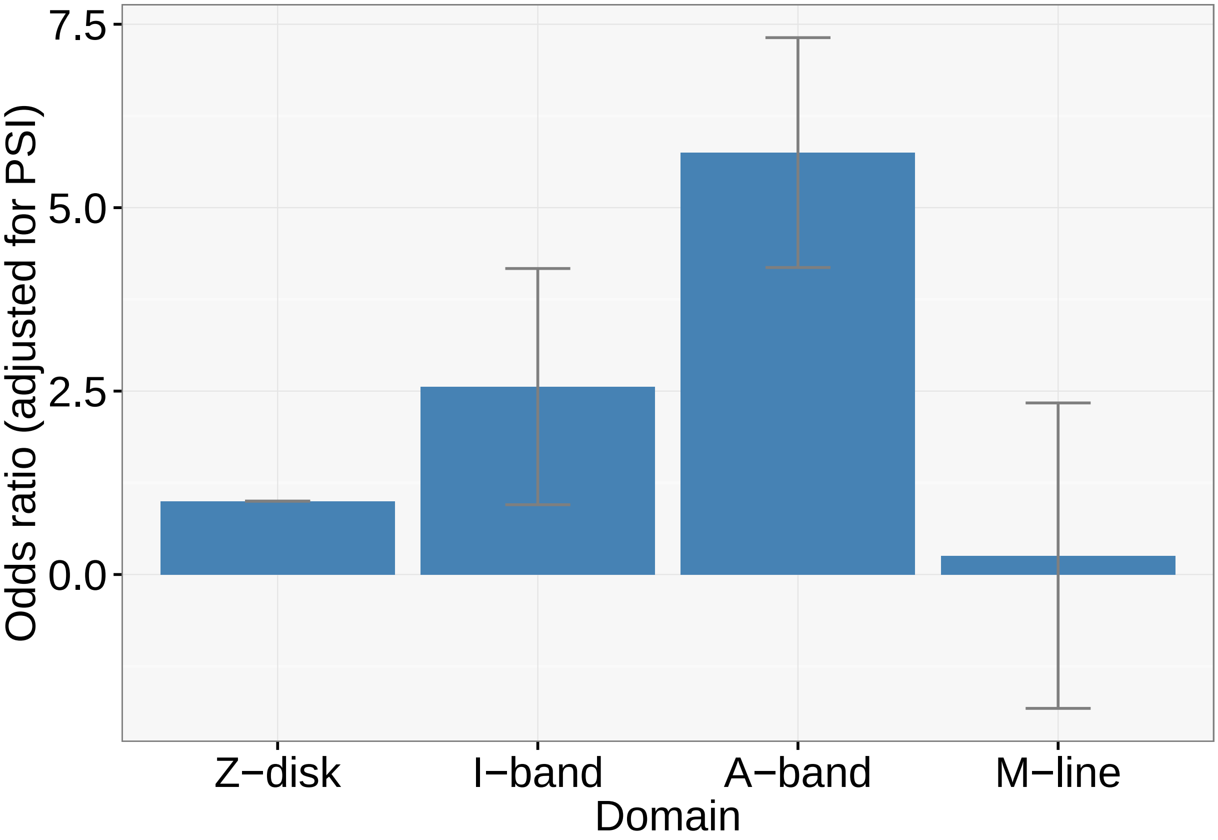
Plot for variation across electron micrograph defined bins, adjusted for splicing

```
model.em <- glm(status ~ domain + psiexp, family = binomial(link = "logit"), data = DCM.CTL.all)
summary(model.em)
```

```
##
## Call:
## glm(formula = status ~ domain + psiexp, family = binomial(link = "logit"),
##     data = DCM.CTL.all)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1146  -0.8057  -0.2169  -0.2169   2.7427
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.6780     0.5955  -7.855 3.99e-15 ***
## domainI-band   0.9403     0.4761   1.975   0.0483 *
## domainA-band   1.7494     0.4488   3.898 9.70e-05 ***
## domainM-line  -1.3592     0.7332  -1.854   0.0638 .
## psiexp1        0.5268     0.8052   0.654   0.5129
## psiexp2        1.0296     0.5842   1.762   0.0780 .
## psiexp3        2.7791     0.4063   6.841 7.89e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1193.12  on 1142  degrees of freedom
## Residual deviance:  904.29  on 1136  degrees of freedom
## AIC: 918.29
##
## Number of Fisher Scoring iterations: 6
```

```
dom.coeff <- c(0, model.em$coeff[2:4])
dom.se <- c(log(0), summary(model.em)$coeff[,2][2:4])
domnames <- c("Z-disk","I-band","A-band","M-line")
dom.all <- cbind.data.frame(domnames, dom.coeff, dom.se)
dom.all$domnames <- factor(dom.all$domnames, levels = domnames)
limits <- aes(ymax = exp(dom.coeff) + exp(dom.se), ymin=exp(dom.coeff) - exp(dom.se))
dodge <- position_dodge(width=0.9)

p <- ggplot(dom.all, aes(x = domnames, y = exp(dom.coeff)))
p <- p + geom_bar(stat = "identity", fill = "steelblue")  + ylab("Odds ratio (adjusted for PSI)") + xlab
p <- p + geom_errorbar(limits, position = position_dodge(width=0.9), width = 0.25, color = "gray50")
p <- p + theme_bw() + scale_fill_brewer(type = "qual", palette = 1)
p<-p +theme(axis.title.x = element_text(size = 16), axis.title.y = element_text(size = 16), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p
```



```
ggsave("comparison_of_case_vs_control_mutation_distribution_electron_micrograph_domains.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

Look at how raw data differs for inclusion or exclusion of expanded CTLs

```
#PSI classes
table(DCM.CTL.all$psiexp)
```

```
##
##   0   1   2   3
## 344  57  80 662
```

```
#AA bins
table(DCM.CTL.all$aabin)
```

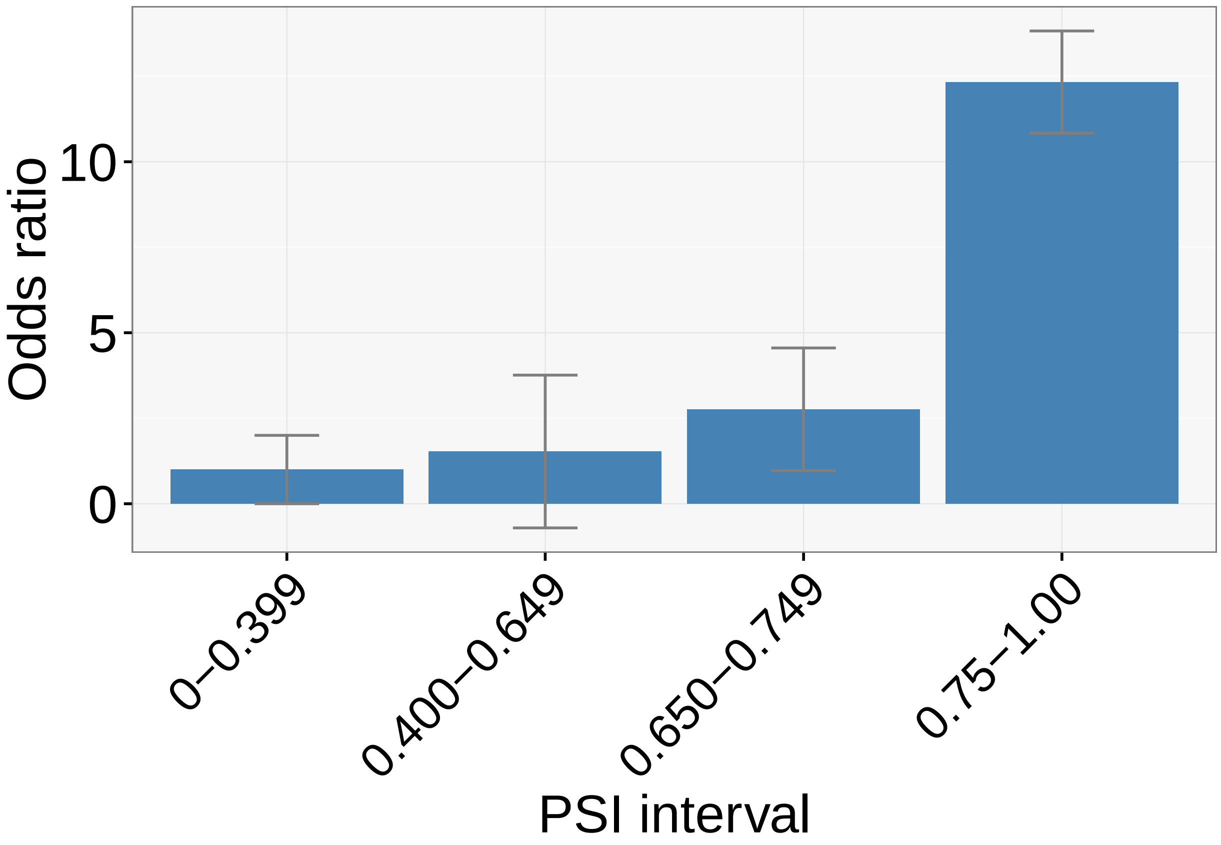
```
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
##  58 187  40  32  83 150  78  43  46  58  39  33  31  45  43  34  30 113
```

Fit an adjusted model for PSI bins along with Cronos position.

```
model.cronos.psi.bin <- glm(status ~ cronos + psiexp, family = binomial(link = "logit"), data = DCM.CTL
summary(model.cronos.psi.bin)
```

```
##
## Call:
## glm(formula = status ~ cronos + psiexp, family = binomial(link = "logit"),
##     data = DCM.CTL.all)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9983  -0.9983  -0.2169  -0.2169   2.7427
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.7377     0.3577 -10.448  < 2e-16 ***
## cronosTRUE    0.7887     0.2055   3.838 0.000124 ***
## psiexp1       0.4235     0.8038   0.527 0.598313
## psiexp2       1.0157     0.5844   1.738 0.082194 .
## psiexp3       2.5120     0.4022   6.245 4.23e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1193.12  on 1142  degrees of freedom
## Residual deviance:  972.41  on 1138  degrees of freedom
## AIC: 982.41
##
## Number of Fisher Scoring iterations: 6
```

```
psi.coeff <- c(0, model.cronos.psi.bin$coeff[3:5])
psi.se <- c(0, summary(model.cronos.psi.bin)$coeff[,2][3:5])
psinames <- c("0-0.399", "0.400-0.649", "0.650-0.749", "0.75-1.00")
psi.all <- cbind.data.frame(psinames, psi.coeff, psi.se)

limits <- aes(ymax = exp(psi.coeff) + exp(psi.se), ymin=exp(psi.coeff) - exp(psi.se))
dodge <- position_dodge(width=0.9)

p <- ggplot(psi.all, aes(x = psinames, y = exp(psi.coeff)))
p <- p + geom_bar(stat = "identity", fill = "steelblue")  + ylab("Odds ratio") + xlab("PSI interval")
p <- p + geom_errorbar(limits, position = position_dodge(width=0.9), width = 0.25, color = "gray50")
p <- p + theme_bw() + scale_fill_brewer(type = "qual", palette = 1)
p<-p +theme(axis.title.x = element_text(size = 20), axis.title.y = element_text(size = 20), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p
```



```
ggsave("comparison_of_DCM_vs_CTL_mutation_distribution_psi_bin.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

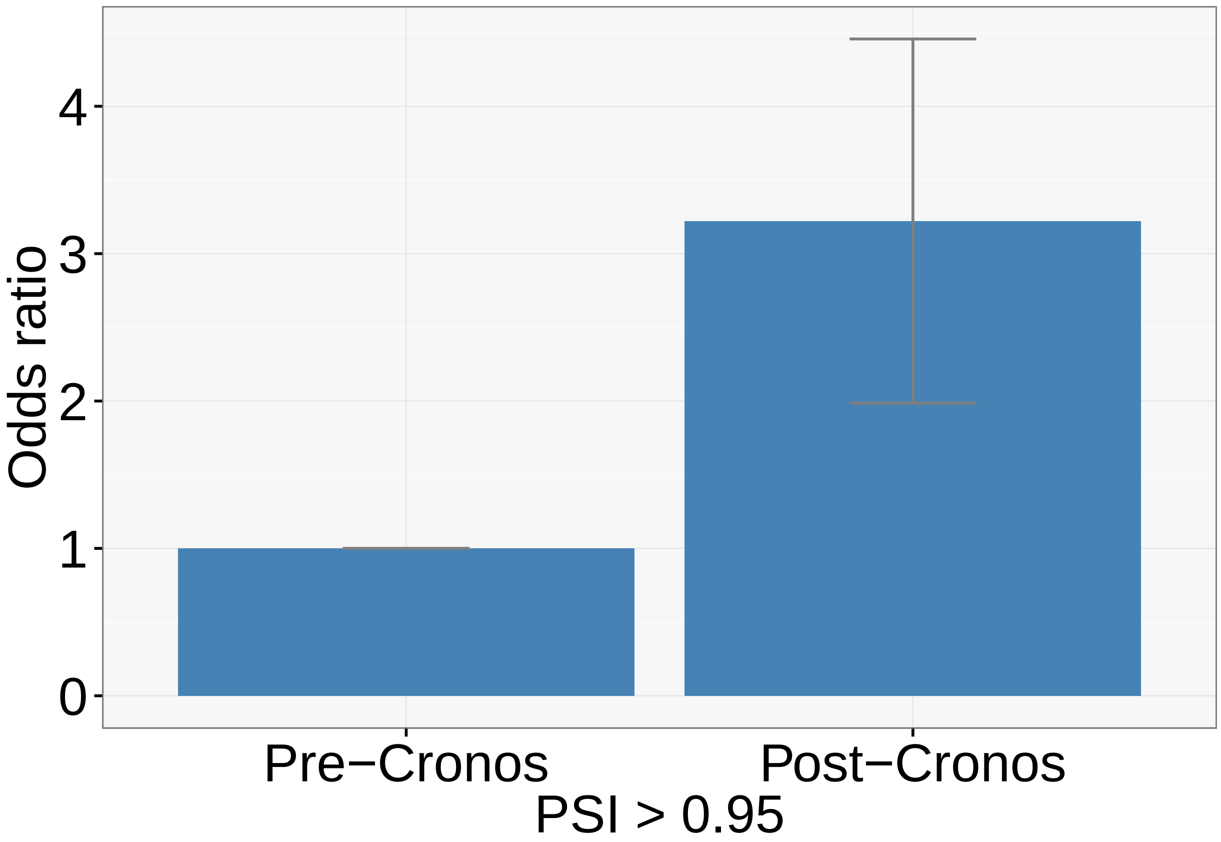
Focus on how the case-control distribution varies whether one is upstream or downstream of Cronos

```
DCM.CTL.all$psiexpgroup <- rep(NA, nrow(DCM.CTL.all))
DCM.CTL.all$psiexpgroup[DCM.CTL.all$psiexp == 0] = "very low"
DCM.CTL.all$psiexpgroup[DCM.CTL.all$psiexp == 1] = "low"
DCM.CTL.all$psiexpgroup[DCM.CTL.all$psiexp == 2] = "medium"
DCM.CTL.all$psiexpgroup[DCM.CTL.all$psiexp == 3] = "high"

DCM.CTL.all$psiexpgroup <- factor(DCM.CTL.all$psiexpgroup, levels = c("very low","low","medium","high")

DCM.CTL.all$Cterm <- rep(NA, nrow(DCM.CTL.all))
DCM.CTL.all$Cterm[DCM.CTL.all$aabin == 17] = "C-term"
DCM.CTL.all$Cterm[!DCM.CTL.all$aabin == 17] = "not C-term"

model.cronos.Cterm <- glm(status ~ cronos + Cterm, family = binomial(link = "logit"), data = DCM.CTL.al
summary(model.cronos.Cterm)
```

```
##
## Call:
## glm(formula = status ~ cronos + Cterm, family = binomial(link = "logit"),
##     data = DCM.CTL.all[DCM.CTL.all$const == TRUE, ])
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1618  -1.1618  -0.3576   1.1931   2.3586
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.8874     0.4438  -8.759  < 2e-16 ***
## cronosTRUE        1.1699     0.2114   5.534 3.14e-08 ***
## Ctermnot C-term   2.6808     0.4035   6.644 3.05e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 854.99  on 658  degrees of freedom
## Residual deviance: 758.63  on 656  degrees of freedom
## AIC: 764.63
##
## Number of Fisher Scoring iterations: 5
```

```
dom.coeff <- c(0, model.cronos.Cterm$coeff[2])
dom.se <- c(log(0), summary(model.cronos.Cterm)$coeff[,2][2])
domnames <- c("Pre-Cronos","Post-Cronos")
dom.all <- cbind.data.frame(domnames, dom.coeff, dom.se)
dom.all$domnames <- factor(dom.all$domnames, levels = domnames)
limits <- aes(ymax = exp(dom.coeff) + exp(dom.se), ymin=exp(dom.coeff) - exp(dom.se))
dodge <- position_dodge(width=0.9)

p <- ggplot(dom.all, aes(x = domnames, y = exp(dom.coeff)))
p <- p + geom_bar(stat = "identity", fill = "steelblue")  + ylab("Odds ratio") + xlab("PSI > 0.95")
p <- p + geom_errorbar(limits, position = position_dodge(width=0.9), width = 0.25, color = "gray50")
p <- p + theme_bw() + scale_fill_brewer(type = "qual", palette = 1)
p<-p +theme(axis.title.x = element_text(size = 20), axis.title.y = element_text(size = 20), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p
```



```
ggsave("comparison_of_case_vs_control_mutation_distribution_constitutive.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

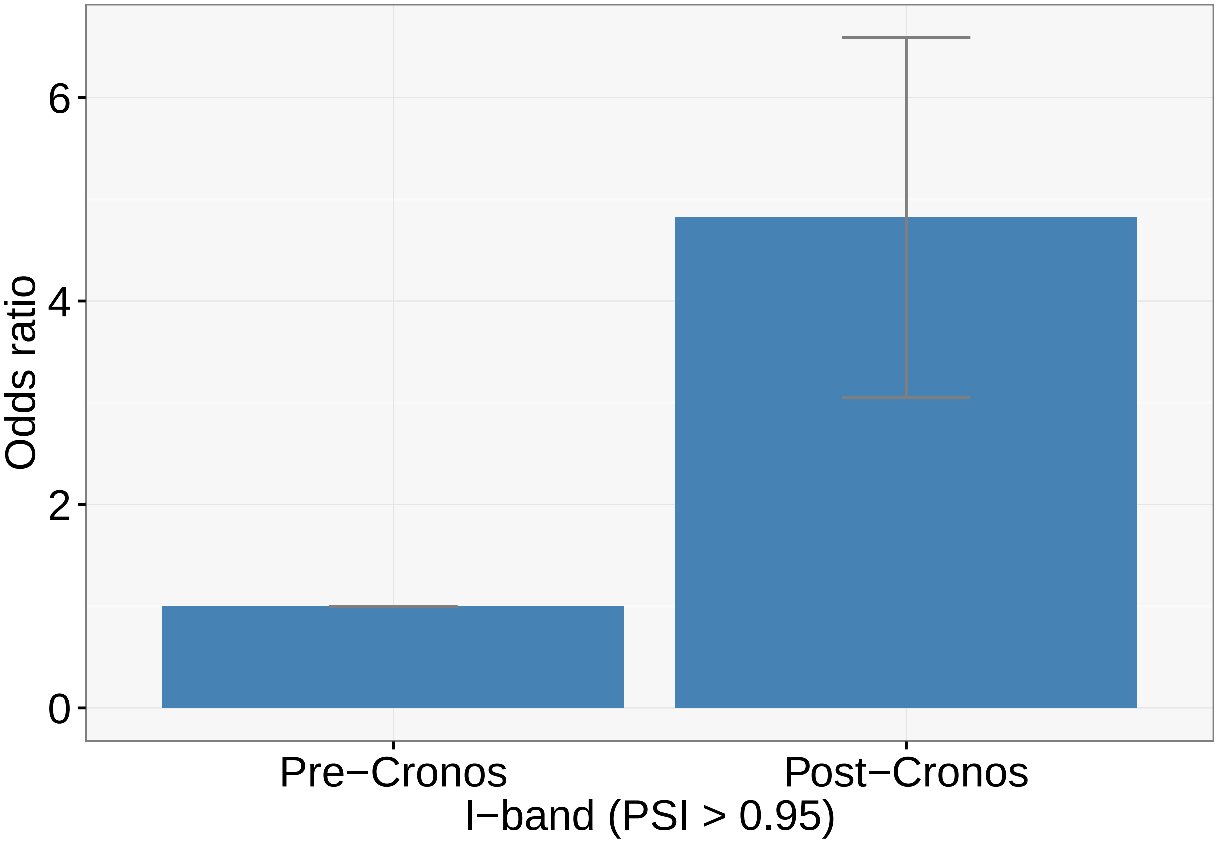
Same plot as above, but just looking at I-band

```
model.Iband <- glm(status ~ cronos , family = binomial(link = "logit"), data = DCM.CTL.all[DCM.CTL.all$
summary(model.Iband)
```

```
##
## Call:
## glm(formula = status ~ cronos, family = binomial(link = "logit"),
##     data = DCM.CTL.all[DCM.CTL.all$const == TRUE & DCM.CTL.all$domain ==
##         "I-band", ])
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.354  -0.736  -0.736   1.011   1.696
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1676     0.2164  -5.396 6.82e-08 ***
## cronosTRUE    1.5731     0.5697   2.761  0.00576 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 157.27  on 132  degrees of freedom
## Residual deviance: 149.50  on 131  degrees of freedom
## AIC: 153.5
##
## Number of Fisher Scoring iterations: 4
```

```
dom.coeff <- c(0, model.Iband$coeff[2])
dom.se <- c(log(0), summary(model.Iband)$coeff[,2][2])
domnames <- c("Pre-Cronos","Post-Cronos")
dom.all <- cbind.data.frame(domnames, dom.coeff, dom.se)
dom.all$domnames <- factor(dom.all$domnames, levels = domnames)
limits <- aes(ymax = exp(dom.coeff) + exp(dom.se), ymin=exp(dom.coeff) - exp(dom.se))
dodge <- position_dodge(width=0.9)

p <- ggplot(dom.all, aes(x = domnames, y = exp(dom.coeff)))
p <- p + geom_bar(stat = "identity", fill = "steelblue")  + ylab("Odds ratio") + xlab("I-band (PSI > 0.
p <- p + geom_errorbar(limits, position = position_dodge(width=0.9), width = 0.25, color = "gray50")
p <- p + theme_bw() + scale_fill_brewer(type = "qual", palette = 1)
p<-p +theme(axis.title.x = element_text(size = 16), axis.title.y = element_text(size = 16), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p
```



```
ggsave("comparison_of_case_vs_control_mutation_distribution_Iband_constitutive.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

Standardize predictors by dividing by standard deviation. This will allow some comparison across predictors.

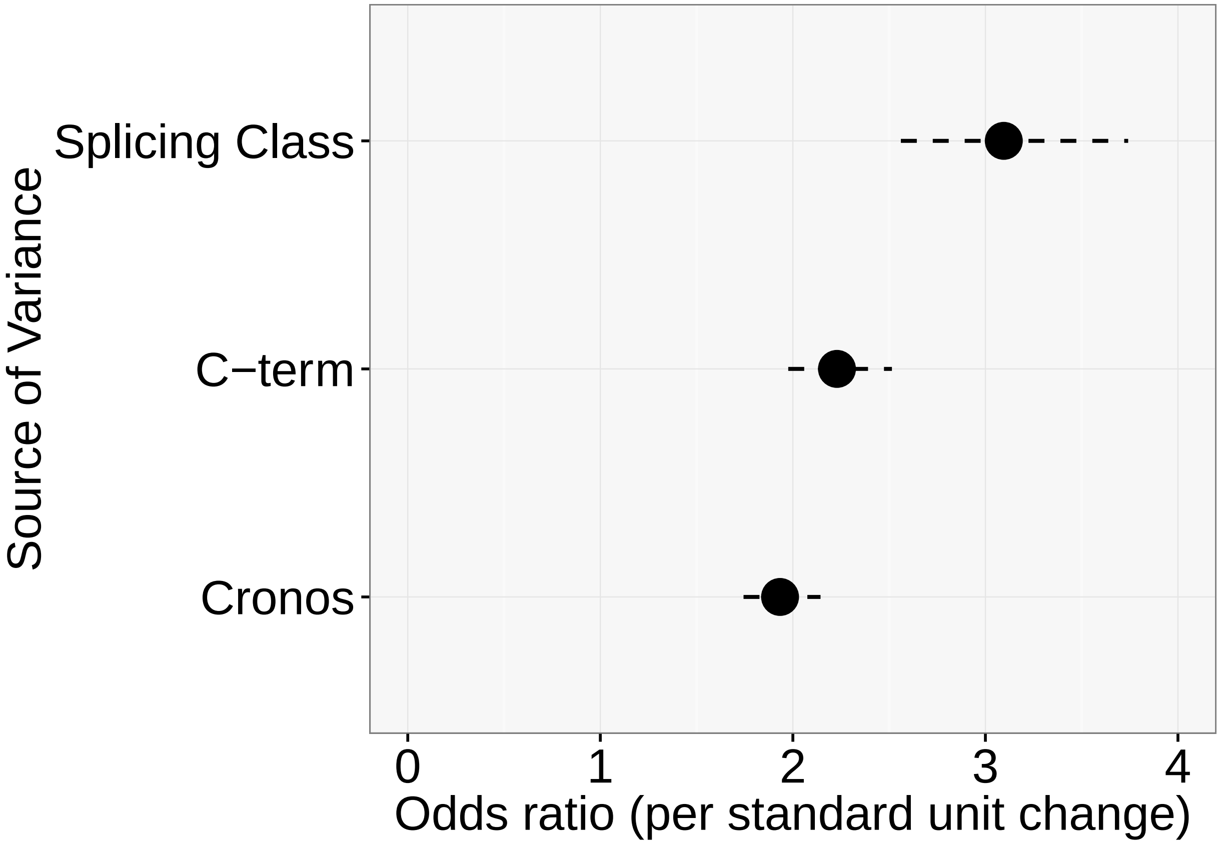
```
DCM.CTL.all$Ctermint <- rep(NA, nrow(DCM.CTL.all))
DCM.CTL.all$Ctermint[DCM.CTL.all$aabin == 17] = 0
DCM.CTL.all$Ctermint[!DCM.CTL.all$aabin == 17] = 1

DCM.CTL.all$cronosint <- rep(NA, nrow(DCM.CTL.all))
DCM.CTL.all$cronosint[DCM.CTL.all$cronos == TRUE] = 1
DCM.CTL.all$cronosint[!DCM.CTL.all$cronos == TRUE] = 0

#we are using odds ratios from the PSI distribution for this step; we will preferably just use PSI as a
DCM.CTL.all$cronosint <- scale(DCM.CTL.all$cronosint, scale = TRUE, center = TRUE)
DCM.CTL.all$psistd <- scale(DCM.CTL.all$psi, scale = TRUE, center = TRUE)
DCM.CTL.all$Ctermint <- scale(DCM.CTL.all$Ctermint, scale = TRUE, center = TRUE)
```

Caterpillar plot

```
model.1 <- glm(status ~ cronosint + Ctermint + psistd, family = binomial(link = "logit"), data = DCM.CTL
summary(model.1)
```

```
##
## Call:
## glm(formula = status ~ cronosint + Ctermint + psistd, family = binomial(link = "logit"),
##     data = DCM.CTL.all)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1650  -0.6762  -0.2608  -0.1863   2.8523
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.9339     0.1326 -14.588  < 2e-16 ***
## cronosint     0.6593     0.1032   6.388 1.68e-10 ***
## Ctermint      0.8016     0.1205   6.654 2.86e-11 ***
## psistd        1.1299     0.1896   5.961 2.51e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1193.1  on 1142  degrees of freedom
## Residual deviance:  900.5  on 1139  degrees of freedom
## AIC: 908.5
##
## Number of Fisher Scoring iterations: 6
```

```
std.coeff <- c(model.1$coeff[2:4])
std.se <- c(summary(model.1)$coeff[,2][2:4])
stdnames <- c("Cronos","C-term","Splicing Class")
std.all <- cbind.data.frame(stdnames, std.coeff, std.se)
std.all$stdnames <- factor(std.all$stdnames, levels = stdnames)
#limits <- aes(ymax = exp(dom.coeff) + exp(dom.se), ymin=exp(dom.coeff) - exp(dom.se))
dodge <- position_dodge(width=0.9)

p <- ggplot(std.all, aes(x = stdnames, y = exp(std.coeff), ymin = exp(std.coeff - std.se), ymax = exp(s
#p <- p + geom_hline(x = 0, linetype = "dotted")
p <- p + geom_pointrange(size = 0.7, linetype = "dashed")
p <- p + geom_point(alpha = 1.0, size = 6)
p <- p + theme_bw() + scale_fill_brewer(type = "qual", palette = 1)
p<-p +theme(axis.title.x = element_text(size = 18), axis.title.y = element_text(size = 18), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p <- p + ylab("Odds ratio (per standard unit change)") + xlab("Source of Variance")
p <- p + ylim(0, 4)
p <- p + coord_flip()
p
```



```
ggsave("caterpillar_TTN_4level.pdf",  useDingbats = FALSE)
```

```
## Saving 6.5 x 4.5 in image
```

Generate predictive model using training set (2/3 of data)

```
data.minim <- DCM.CTL.all[,c(16,17,21,22)]
model.1 <- glm(status ~ cronos + Cterm + psiexpgroup, family = binomial(link = "logit"), data = data.mi
summary(model.1)
```

```
##
## Call:
## glm(formula = status ~ cronos + Cterm + psiexpgroup, family = binomial(link = "logit"),
##     data = data.minim)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1611  -0.7180  -0.2169  -0.2169   2.7427
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -6.4166     0.5392 -11.900  < 2e-16 ***
## cronosTRUE           1.1854     0.2101   5.643 1.67e-08 ***
## Ctermnot C-term      2.6789     0.4035   6.640 3.14e-11 ***
## psiexpgrouplow       0.4235     0.8038   0.527   0.5983
## psiexpgroupmedium    1.0049     0.5848   1.719   0.0857 .
## psiexpgrouphigh      2.5137     0.4021   6.251 4.08e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1193.12  on 1142  degrees of freedom
## Residual deviance:  891.17  on 1137  degrees of freedom
## AIC: 903.17
##
## Number of Fisher Scoring iterations: 6
```

```
niter = 100

auroc <- rep(NA, niter)

library(ROCR)
```

```
## Loading required package: gplots
```

```
## Warning: package 'gplots' was built under R version 3.2.4
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
for (i in 1:niter)
{
train <- sample(1:nrow(DCM.CTL.all), round(0.66*nrow(DCM.CTL.all)))
data.train <- data.minim[c(train),]
data.test <- data.minim[-c(train),]
model.1 <- glm(status ~ cronos + Cterm + psiexpgroup , family = binomial(link = "logit"), data = data.t
model.test <- predict.glm(model.1, data.test)
preds <- prediction(model.test, data.test$status)
perf <- performance(preds, "auc")
auroc[i] = as.numeric(perf@y.values[[1]])
}
print(summary(auroc))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7769  0.8133  0.8246  0.8231  0.8338  0.8617
```

```
auroc.psiexp <- auroc
perf <- performance(preds, "tpr","fpr")
print(perf)
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "False positive rate"
##
## Slot "y.name":
## [1] "True positive rate"
##
## Slot "alpha.name":
## [1] "Cutoff"
##
## Slot "x.values":
## [[1]]
## [1] 0.0000000 0.2150171 0.3378840 0.3412969 0.4266212 0.5392491 0.6075085
## [8] 1.0000000
##
##
## Slot "y.values":
## [[1]]
## [1] 0.0000000 0.7500000 0.9062500 0.9062500 0.9270833 0.9583333 0.9687500
## [8] 1.0000000
##
##
## Slot "alpha.values":
## [[1]]
## [1]        Inf -0.1302652 -1.4078006 -1.5156726 -2.7932080 -2.9041651
## [7] -3.5553481 -3.7887248
```

```
table(attributes(preds)$predictions[[1]])
```

```
##
##   -3.7887247873636  -3.5553480614546 -2.90416508002855
##                118                21                36
##  -2.79320800944256  -1.51567263793963   -1.4078005663409
##                 27                  1                 51
## -0.130265194837965
##                135
```

```
#Look at discrete bins of patients
print(table(data.minim))
```

```
## , , psiexpgroup = very low, Cterm = C-term
## 
##       cronos
## status FALSE TRUE
##    CTL     0    0
##    DCM     0    0
## 
## , , psiexpgroup = low, Cterm = C-term
## 
##       cronos
## status FALSE TRUE
##    CTL     0    0
##    DCM     0    0
## 
## , , psiexpgroup = medium, Cterm = C-term
## 
##       cronos
## status FALSE TRUE
##    CTL     0    0
##    DCM     0    0
## 
## , , psiexpgroup = high, Cterm = C-term
## 
##       cronos
## status FALSE TRUE
##    CTL     0  106
##    DCM     0    7
## 
## , , psiexpgroup = very low, Cterm = not C-term
## 
##       cronos
## status FALSE TRUE
##    CTL   336    0
##    DCM     8    0
## 
## , , psiexpgroup = low, Cterm = not C-term
## 
##       cronos
## status FALSE TRUE
##    CTL    55    0
##    DCM     2    0
## 
## , , psiexpgroup = medium, Cterm = not C-term
## 
##       cronos
## status FALSE TRUE
##    CTL    74    1
##    DCM     5    0
##
## , , psiexpgroup = high, Cterm = not C-term
##
##       cronos
## status FALSE TRUE
##    CTL   130  194
##    DCM    38  187
```

```
#Consider PSI as a continuous variable
data.minim <- DCM.CTL.all[,c(16,17,21,22,14)]
model.1 <- glm(status ~ cronos + Cterm + psi, family = binomial(link = "logit"), data = data.minim)
summary(model.1)
```

```
##
## Call:
## glm(formula = status ~ cronos + Cterm + psi, family = binomial(link = "logit"),
##     data = data.minim)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1650  -0.6762  -0.2608  -0.1863   2.8523
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -6.7405     0.5537 -12.173  < 2e-16 ***
## cronosTRUE         1.3300     0.2082   6.388 1.68e-10 ***
## Ctermnot C-term    2.6844     0.4035   6.654 2.86e-11 ***
## psi                2.6969     0.4524   5.961 2.51e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1193.1  on 1142  degrees of freedom
## Residual deviance:  900.5  on 1139  degrees of freedom
## AIC: 908.5
##
## Number of Fisher Scoring iterations: 6
```

```
for (i in 1:niter)
{
train <- sample(1:nrow(DCM.CTL.all), round(0.66*nrow(DCM.CTL.all)))
data.train <- data.minim[c(train),]
data.test <- data.minim[-c(train),]
model.1 <- glm(status ~ cronos + Cterm + psi , family = binomial(link = "logit"), data = data.train)
model.test <- predict.glm(model.1, data.test)
preds <- prediction(model.test, data.test$status)
perf <- performance(preds, "auc")
auroc[i] = as.numeric(perf@y.values[[1]])
}
print(summary(auroc))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7549  0.7924  0.8035  0.8053  0.8176  0.8548
```

```
auroc.psicont <- auroc
```

Set the C-terminal threshold at the end of the kinase domain and repeat AUROC analysis.

```
Ctermthresh = 34092

DCM.CTL.all$Ctermkin <- rep(0, nrow(DCM.CTL.all))
DCM.CTL.all$Ctermkin[DCM.CTL.all$aa_map < Ctermthresh] = 1

data.minim <- DCM.CTL.all[,c(16,17,21,26)]
model.1 <- glm(status ~ cronos + Ctermkin + psiexpgroup, family = binomial(link = "logit"), data = data
summary(model.1)
```

```
##
## Call:
## glm(formula = status ~ cronos + Ctermkin + psiexpgroup, family = binomial(link = "logit"),
##     data = data.minim)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1546  -0.7180  -0.2169  -0.2169   2.7427
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -6.3726     0.5394 -11.814  < 2e-16 ***
## cronosTRUE          1.1702     0.2099   5.575 2.47e-08 ***
## Ctermkin            2.6349     0.4037   6.527 6.71e-11 ***
## psiexpgrouplow      0.4235     0.8038   0.527   0.5983
## psiexpgroupmedium   1.0054     0.5847   1.719   0.0855 .
## psiexpgrouphigh     2.5136     0.4021   6.251 4.09e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1193.12  on 1142  degrees of freedom
## Residual deviance:  894.81  on 1137  degrees of freedom
## AIC: 906.81
##
## Number of Fisher Scoring iterations: 6
```

```
niter = 100

auroc <- rep(NA, niter)

library(ROCR)
for (i in 1:niter)
{
train <- sample(1:nrow(DCM.CTL.all), round(0.66*nrow(DCM.CTL.all)))
data.train <- data.minim[c(train),]
data.test <- data.minim[-c(train),]
model.1 <- glm(status ~ cronos + Ctermkin + psiexpgroup, family = binomial(link = "logit"), data = data
model.test <- predict.glm(model.1, data.test)
preds <- prediction(model.test, data.test$status)
perf <- performance(preds, "auc")
auroc[i] = as.numeric(perf@y.values[[1]])
}
print(summary(auroc))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7659  0.8059  0.8198  0.8171  0.8309  0.8545
```

```
auroc.psiexp <- auroc
perf <- performance(preds, "tpr","fpr")
print(perf)
```

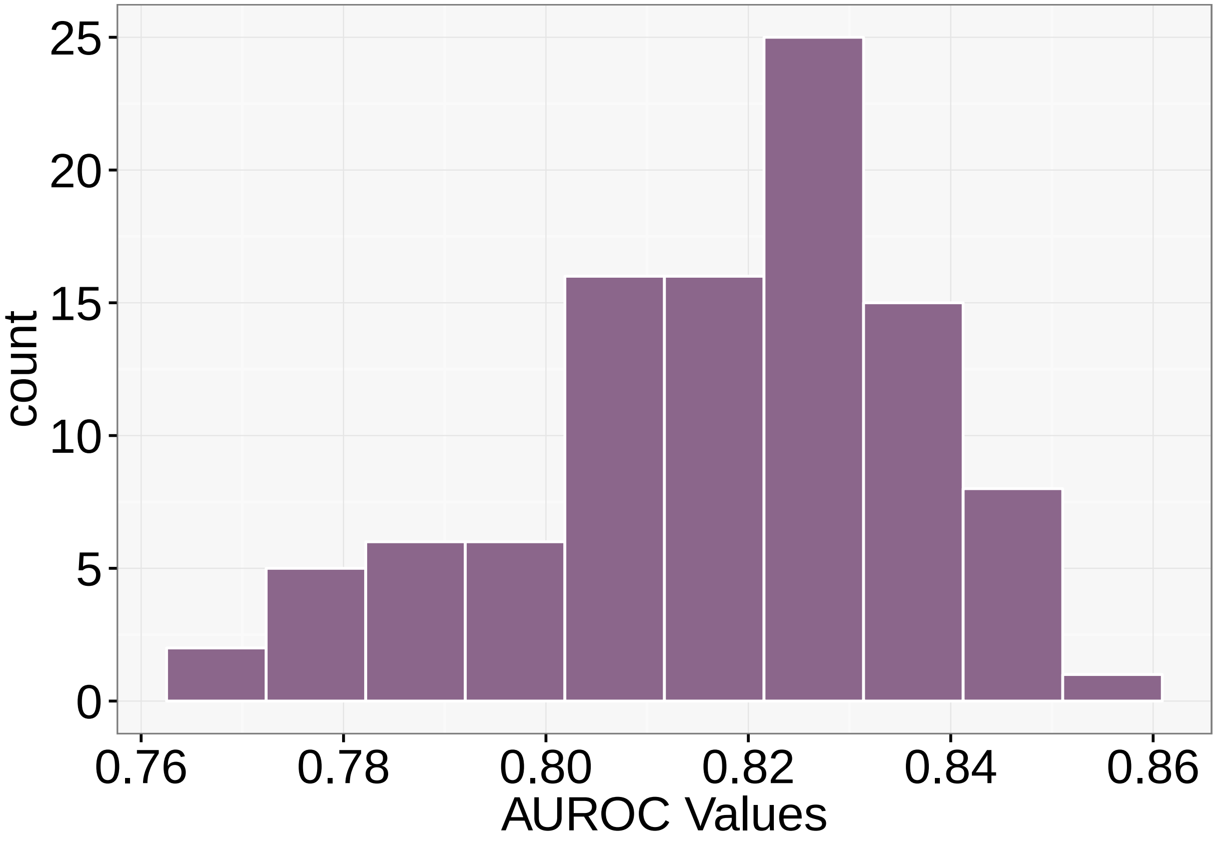
```
## An object of class "performance"
## Slot "x.name":
## [1] "False positive rate"
##
## Slot "y.name":
## [1] "True positive rate"
##
## Slot "alpha.name":
## [1] "Cutoff"
##
## Slot "x.values":
## [[1]]
## [1] 0.0000000 0.2013423 0.3456376 0.4395973 0.5536913 0.6140940 1.0000000
##
##
## Slot "y.values":
## [[1]]
## [1] 0.0000000 0.7912088 0.9120879 0.9120879 0.9450549 0.9560440 1.0000000
##
##
## Slot "alpha.values":
## [[1]]
## [1]        Inf -0.1785099 -1.1596001 -2.2681307 -2.8478121 -3.6109179
## [7] -4.0118683
```

```
table(attributes(preds)$predictions[[1]])
```

```
##
##   -4.0118683015189  -3.61091791254993  -2.84781214347739
##                119                 19                 37
##  -2.26813073789844  -1.15960014340769 -0.178509891715003
##                 28                 54                132
```

Plot distribution of AUROC values

```
auroc.data <- data.frame(auroc)
p <- ggplot(data = auroc.data, aes(x = auroc))
p <- p + geom_histogram(fill = "plum4", bins=10, colour = "white")
p <- p + theme_bw()
p<-p +theme(axis.title.x = element_text(size = 18), axis.title.y = element_text(size = 18), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p <- p +  xlab("AUROC Values")
p
```



```
ggsave("auroc_distribution.pdf",  useDingbats = FALSE)
```

```
## Saving 6.5 x 4.5 in image
```

Plot ROC

```
library(verification)
```

```
## Loading required package: fields
```

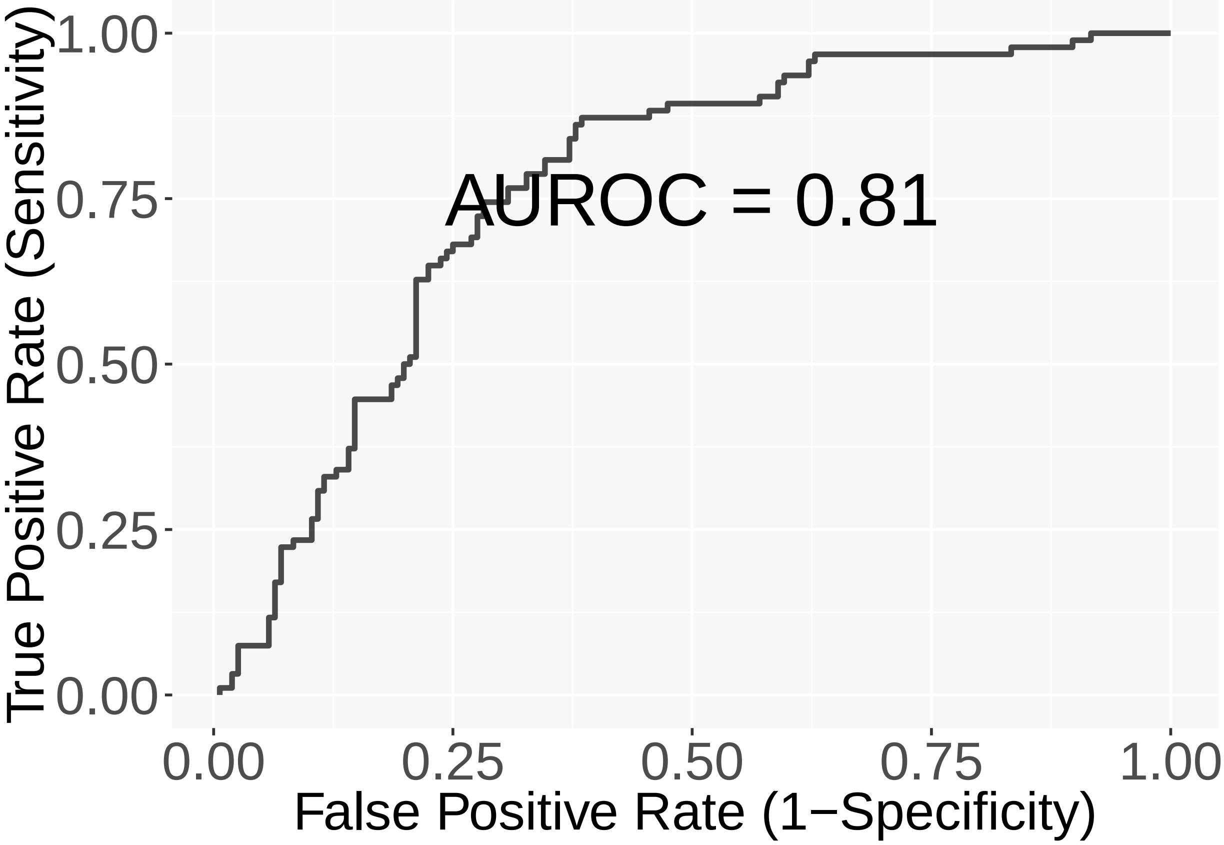
```
## Warning: package 'fields' was built under R version 3.2.5
```

```
## Loading required package: spam
```

```
## Loading required package: grid
## Spam version 1.3-0 (2015-10-24) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: maps

##
##  # maps v3.1: updated 'world': all lakes moved to separate new #
##  # 'lakes' database. Type '?world' or 'news(package="maps")'.  #

## Loading required package: boot

## Loading required package: CircStats

## Loading required package: MASS

## Loading required package: dtw

## Loading required package: proxy

## Warning: package 'proxy' was built under R version 3.2.5

##
## Attaching package: 'proxy'

## The following object is masked from 'package:spam':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix

## Loaded dtw v1.18-1. See ?dtw for help, citation("dtw") for use in publication.
```

```
data.minim <- DCM.CTL.all[,c(16,17,21,26)]
train <- sample(1:nrow(DCM.CTL.all), round(0.66*nrow(DCM.CTL.all)))
data.train <- data.minim[c(train),]
data.test <- data.minim[-c(train),]
model.1 <- glm(status ~ cronos + Ctermkin + psiexpgroup, family = binomial(link = "logit"), data = data
model.test <- predict.glm(model.1, data.test)
preds <- prediction(model.test, data.test$status)
perf <- performance(preds, "auc")
perf.tpr <- performance(preds, "tpr","fpr")

ROCdata <- data.frame("pos" = data.test$status, "annotated" = model.test)

basal <- ROCdata
basal <- basal[order(basal[,2], decreasing = TRUE),]

l = length(unique(basal[,2]))
#scramble
basal.1 <- basal[cut(basal[,2], breaks = l) == levels(cut(basal[,2], breaks = l))[l],]
basal.2 <- basal[cut(basal[,2], breaks = l) == levels(cut(basal[,2], breaks = l))[l-1],]
basal.3 <- basal[cut(basal[,2], breaks = l) == levels(cut(basal[,2], breaks = l))[l-2],]
basal.4 <- basal[cut(basal[,2], breaks = l) == levels(cut(basal[,2], breaks = l))[l-3],]
basal.5 <- basal[cut(basal[,2], breaks = l) == levels(cut(basal[,2], breaks = l))[l-4],]
basal.6 <- basal[cut(basal[,2], breaks = l) == levels(cut(basal[,2], breaks = l))[l-5],]

basal.1 <- basal.1[sample(nrow(basal.1)),]
basal.2 <- basal.2[sample(nrow(basal.2)),]
basal.3 <- basal.3[sample(nrow(basal.3)),]
basal.4 <- basal.4[sample(nrow(basal.4)),]
basal.5 <- basal.5[sample(nrow(basal.5)),]
basal.6 <- basal.6[sample(nrow(basal.6)),]

basal <- rbind.data.frame(basal.1, basal.2, basal.3, basal.4, basal.5, basal.6)

  tp <- vector(); tn <-vector(); fp <-vector(); fn <- vector()
  tpr <- vector(); fpr <- vector()
  acc <- vector(); spc <- vector()
  len <- dim(basal)[1]
  for(i in 1:len-1) {
    fn[i] <- sum(basal[(i+1):len,1] == "DCM")
    fp[i] <- sum(basal[1:i,1] == "CTL")
    tn[i] <- sum(basal[(i+1):len,1] == "CTL")
    tp[i] <- sum(basal[1:i,1] == "DCM")
    tpr[i] <- tp[i] / (tp[i] + fn[i])
    fpr[i] <- fp[i] / (fp[i] + tn[i])
    acc[i] <- (tp[i] + tn[i]) / ((tp[i] + fn[i]) +  (fp[i] + tn[i]))
    spc[i] <- 1 - fpr[i]
  }
  points <- (cbind(fpr,tpr))#[(len-1):1,]
  points <- rbind(points, c(1,1))

xlabel = "False Positive Rate (1-Specificity)"
ylabel = "True Positive Rate (Sensitivity)"
```

```r
data <- data.frame(TPR = points[,2], FPR = points[,1])
p <- ggplot(data,aes(x=FPR, y=TPR)) + xlab(xlabel) + ylab(ylabel)
p <- p+geom_line(size=1, alpha=0.7)
p<-p +theme(axis.title.x = element_text(size = 20), axis.title.y = element_text(size = 20), axis.text.x
p <- p + theme(panel.background = element_rect(fill = 'gray97'))
p <- p + annotate("text", label = "AUROC = 0.81", x = 0.5, y = 0.75, size = 10)
p
```



```r
ggsave("ttn_ROC_psi_discrete_keep.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

Bin patients into 6 bins based on: 1. Splicing extent: very low, low, medium high 2. Cronos disruption: yes or no 3. C-term not disrupted: yes or no

Although this gives a maximum of 2 x 2 x 4 = 16 categories, only 6 of these have more than 1 individual (9 bins are empty).

Compute Fisher's Exact Test, significance, and confidence intervals. The null hypothesis is that belonging to any given bin has no impact on your probability of having a TTN truncating variant

```r
#total patients 639 Haas, Roberts (End stage 155, Unselected DCM, 371 unrelated, replication 163), Herm

DCMtotal = 639 + 155 + 371 + 163 + 312 -71 + 145 # (1714)
TTNDCMtotal = nrow(DCM.all.rep)
```

```
TTNCTLtotal = nrow(CTL.all.rep)

#Controls

# 60,706 (ExAC), 2504 (1000G), 6000 (EVS):  69210
CTLtotal = 60706 + 2504 + 6000

extractfisher <- function(CTLindex, DCMindex, datatable)
{
  a = datatable
  b = fisher.test(matrix(c(a[DCMindex], a[CTLindex],DCMtotal - a[DCMindex], CTLtotal - a[CTLindex]), nr
  pval <- b$p.value
  OR <- round(b$estimate,1)
  CI <- round(b$conf.int,1)
  round(fracDCMTTN <- a[DCMindex]/TTNDCMtotal,3)
  round(fracCTLTTN <- a[CTLindex]/TTNCTLtotal,3)
  round(fracDCM <- a[DCMindex]/DCMtotal,3)
  round(fracCTL <- a[CTLindex]/CTLtotal,3)
  out <- c(pval, OR, CI, fracDCMTTN, fracCTLTTN, fracDCM, fracCTL)
  names(out) <- c("pvalue","OR","95% CI lower", "95% CI upper", "DCM TTN fraction", "CTL TTN fraction",
  out;
}

a = table(data.minim)
print(a)
```

```
## , , psiexpgroup = very low, Ctermkin = 0
##
##       cronos
## status FALSE TRUE
##    CTL     0    0
##    DCM     0    0
##
## , , psiexpgroup = low, Ctermkin = 0
##
##       cronos
## status FALSE TRUE
##    CTL     0    0
##    DCM     0    0
##
## , , psiexpgroup = medium, Ctermkin = 0
##
##       cronos
## status FALSE TRUE
##    CTL     0    0
##    DCM     0    0
##
## , , psiexpgroup = high, Ctermkin = 0
##
##       cronos
## status FALSE TRUE
##    CTL     0  103
##    DCM     0    7
```

```
##
## , , psiexpgroup = very low, Ctermkin = 1
##
##       cronos
## status FALSE TRUE
##    CTL   336    0
##    DCM     8    0
##
## , , psiexpgroup = low, Ctermkin = 1
##
##       cronos
## status FALSE TRUE
##    CTL    55    0
##    DCM     2    0
##
## , , psiexpgroup = medium, Ctermkin = 1
##
##       cronos
## status FALSE TRUE
##    CTL    74    1
##    DCM     5    0
##
## , , psiexpgroup = high, Ctermkin = 1
##
##       cronos
## status FALSE TRUE
##    CTL   130  197
##    DCM    38  187
```

```
#very low PSI, Cterm, no Cronos:
print(extractfisher(1,2,a), digits = 2)
```

```
##           pvalue                OR     95% CI lower     95% CI upper
##                1                 0                0              Inf
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction
##                0                 0                0                0
```

```
#very low PSI, Cterm, yes Cronos:
print(extractfisher(3,4,a), digits = 2)
```

```
##           pvalue                OR     95% CI lower     95% CI upper
##                1                 0                0              Inf
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction
##                0                 0                0                0
```

```
#low PSI, Cterm, no Cronos:
print(extractfisher(5,6,a), digits = 2)
```

```
##           pvalue                OR     95% CI lower     95% CI upper
##                1                 0                0              Inf
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction
##                0                 0                0                0
```

```r
#low PSI, Cterm, yes Cronos:
print(extractfisher(7,8,a), digits = 2)
```

```
##           pvalue               OR     95% CI lower     95% CI upper 
##                1                0                0              Inf 
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction 
##                0                0                0                0
```

```r
#medium PSI, Cterm, no Cronos:
print(extractfisher(9,10,a), digits = 2)
```

```
##           pvalue               OR     95% CI lower     95% CI upper 
##                1                0                0              Inf 
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction 
##                0                0                0                0
```

```r
#medium PSI, Cterm, yes Cronos:
print(extractfisher(11,12,a), digits = 2)
```

```
##           pvalue               OR     95% CI lower     95% CI upper 
##                1                0                0              Inf 
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction 
##                0                0                0                0
```

```r
#high PSI, Cterm, no Cronos:
print(extractfisher(13,14,a), digits = 2)
```

```
##           pvalue               OR     95% CI lower     95% CI upper 
##                1                0                0              Inf 
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction 
##                0                0                0                0
```

```r
#high PSI, Cterm, yes Cronos:
print(extractfisher(15,16,a), digits = 2)
```

```
##           pvalue               OR     95% CI lower     95% CI upper 
##           0.0176           2.8000           1.1000           5.9000 
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction 
##           0.0283           0.1150           0.0041           0.0015
```

```r
#very low PSI, not Cterm, no Cronos:
print(extractfisher(17,18,a), digits = 2)
```

```
##           pvalue               OR     95% CI lower     95% CI upper 
##           1.0000           1.0000           0.4000           1.9000 
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction 
##           0.0324           0.3750           0.0047           0.0049
```

```
#very low PSI, not Cterm, yes Cronos:
print(extractfisher(19,20,a), digits = 2)
```

```
##            pvalue               OR     95% CI lower     95% CI upper
##                 1                0                0              Inf
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction
##                 0                0                0                0
```

```
#low PSI, not Cterm, no Cronos:
print(extractfisher(21,22,a), digits = 2)
```

```
##            pvalue               OR     95% CI lower     95% CI upper
##           0.40203          1.50000          0.20000          5.60000
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction
##           0.00810          0.06138          0.00117          0.00079
```

```
#low PSI, not Cterm, yes Cronos:
print(extractfisher(23,24,a), digits = 2)
```

```
##            pvalue               OR     95% CI lower     95% CI upper
##                 1                0                0              Inf
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction
##                 0                0                0                0
```

```
#medium PSI, not Cterm, no Cronos:
print(extractfisher(25,26,a), digits = 2)
```

```
##            pvalue               OR     95% CI lower     95% CI upper
##            0.0426           2.7000           0.9000           6.7000
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction
##            0.0202           0.0826           0.0029           0.0011
```

```
#medium PSI, not Cterm, yes Cronos:
print(extractfisher(27,28,a), digits = 2)
```

```
##            pvalue               OR     95% CI lower     95% CI upper
##           1.0e+00          0.0e+00          0.0e+00          1.5e+03
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction
##           0.0e+00          1.1e-03          0.0e+00          1.4e-05
```

```
#high PSI, not Cterm, no Cronos:
print(extractfisher(29,30,a), digits = 2)
```

```
##            pvalue               OR     95% CI lower     95% CI upper
##           8.8e-26          1.2e+01          8.1e+00          1.8e+01
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction
##           1.5e-01          1.5e-01          2.2e-02          1.9e-03
```

```r
#high PSI, not Cterm, yes Cronos:
print(extractfisher(31,32,a), digits = 2)
```

```
##           pvalue               OR     95% CI lower     95% CI upper
##         3.0e-195          4.3e+01          3.5e+01          5.3e+01
## DCM TTN fraction CTL TTN fraction     DCM fraction     CTL fraction
##          7.6e-01          2.2e-01          1.1e-01          2.8e-03
```