DSN 2019

Daniel

November 17, 2018

Loading data and libraries

Load ggplot2 for plotting data and the CSV file with all data. We also correct the cross section of thermals since the rotax spectrum is different, which means we should correct the fluence we had before by a factor of (4.96/2.71). Thus, we divide the cross section by this factor since the cross section is (number of errors)/(fluence).

```
library(ggplot2)

dat = read.csv("data_formatted.csv",sep=",")

dat$cross_section_SDC_thermals = dat$cross_section_SDC_thermals / (4.96/2.71)

summary(dat)
```

```
##          Device       Code        Input      cross_section_SDC_high
##  CPU        :3   LavaMD :4   Min.   :   0   Min.   :7.100e-10
##  CPU+GPU    :3   MxM    :4   1st Qu.:   0   1st Qu.:4.700e-09
##  GPUembedded:3   BFS    :3   Median :  20   Median :7.905e-09
##  K20        :4   CED    :3   Mean   :1438   Mean   :1.686e-08
##  TitanV     :2   HotSpot:3   3rd Qu.:1024   3rd Qu.:1.382e-08
##  TitanX     :3   SC     :3   Max.   :8096   Max.   :1.170e-07
##  XeonPhi    :4   (Other):2
##  number_SDCs_high cross_section_DUE_high number_DUE_high
##  Min.   :  8.00   Min.   :1.730e-10      Min.   : 1.00
##  1st Qu.: 28.25   1st Qu.:1.450e-09      1st Qu.: 9.50
##  Median : 48.50   Median :3.410e-09      Median :24.50
##  Mean   : 63.64   Mean   :4.356e-09      Mean   :26.32
##  3rd Qu.: 95.25   3rd Qu.:5.307e-09      3rd Qu.:40.75
##  Max.   :162.00   Max.   :1.520e-08      Max.   :77.00
##
##  cross_section_SDC_thermals number_SDCs_thermals
##  Min.   :2.743e-10          Min.   :  3.00
##  1st Qu.:9.193e-10          1st Qu.: 18.00
##  Median :1.421e-09          Median : 47.00
##  Mean   :6.465e-09          Mean   : 55.73
##  3rd Qu.:6.980e-09          3rd Qu.: 67.25
##  Max.   :3.590e-08          Max.   :202.00
##
##  cross_section_DUE_thermals number_DUE_thermals
##  Min.   :0.000e+00          Min.   : 0.00
##  1st Qu.:4.132e-10          1st Qu.: 6.50
##  Median :1.815e-09          Median :10.00
##  Mean   :2.093e-09          Mean   :15.91
##  3rd Qu.:2.553e-09          3rd Qu.:24.75
##  Max.   :6.250e-09          Max.   :56.00
##
```

Normalize SDC and DUE Cross Sections

Find the minimum value for each device and normalize SDC and DUE cross sections. Then, append the values normalize to a new data frame called "datNormalized"

```
datNormalized = dat
for (board in unique(dat$Device)) {
  # Get all cross section values for one device
  tmp_values = c(subset(dat,dat$Device==board)$cross_section_SDC_high, subset(dat,dat$Device==board)$cro
  tmp_values = tmp_values[tmp_values > 0] # remove zeros in case there is zeros
  tmp_min = min(tmp_values) # get the minimum cross section value
  print(paste("Device: ", board, "; Minimum: ",tmp_min))

  # Normalize all date from one device to the minimum value found
  datNormalized[datNormalized$Device == board, ]$cross_section_SDC_high = datNormalized[datNormalized$D
  datNormalized[datNormalized$Device == board, ]$cross_section_DUE_high = datNormalized[datNormalized$D
  datNormalized[datNormalized$Device == board, ]$cross_section_SDC_thermals = datNormalized[datNormaliz
  datNormalized[datNormalized$Device == board, ]$cross_section_DUE_thermals = datNormalized[datNormaliz
}
```

```
## [1] "Device:  XeonPhi ; Minimum:  3.17e-10"
## [1] "Device:  K20 ; Minimum:  1.73e-10"
## [1] "Device:  TitanV ; Minimum:  5.22e-10"
## [1] "Device:  TitanX ; Minimum:  1.93e-10"
## [1] "Device:  GPUembedded ; Minimum:  3.98850806451613e-10"
## [1] "Device:  CPU ; Minimum:  7.52e-10"
## [1] "Device:  CPU+GPU ; Minimum:  2.74278225806452e-10"
```

Compute FIT for High Energy

Normal flux is 13 neutrons per cm^2 per hour. Since data is normalized, there is no need to multiply by 10^9

```
datNormalized$FIT_SDC_high = datNormalized$cross_section_SDC_high * 13
datNormalized$FIT_DUE_high = datNormalized$cross_section_DUE_high * 13
```

Compute FIT for Thermals

Normal flux is 4 neutrons per cm^2 per hour under normal environments.

```
datNormalized$FIT_SDC_thermals = datNormalized$cross_section_SDC_thermals * 4
datNormalized$FIT_DUE_thermals = datNormalized$cross_section_DUE_thermals * 4
```

During a thunderstrom the neutron flux is about 8 neutrons

```
datNormalized$FIT_SDC_thermals_storm = datNormalized$cross_section_SDC_thermals * 8
datNormalized$FIT_DUE_thermals_storm = datNormalized$cross_section_DUE_thermals * 8
```

Compute the error bars

Using the Poisson distribution with 95% confidence, which is 1.96*sqrt(test_mean/#tests)

```
  datNormalized$cross_section_SDC_high_err = 1.96*sqrt(datNormalized$cross_section_SDC_high/datNormalize
  datNormalized$cross_section_DUE_high_err = 1.96*sqrt(datNormalized$cross_section_DUE_high/datNormalize
  datNormalized$cross_section_SDC_thermals_err = 1.96*sqrt(datNormalized$cross_section_SDC_thermals/datN
  datNormalized$cross_section_DUE_thermals_err = 1.96*sqrt(datNormalized$cross_section_DUE_thermals/datN

  datNormalized$FIT_SDC_high_err = 1.96*sqrt(datNormalized$FIT_SDC_high/datNormalized$number_SDCs_high)
  datNormalized$FIT_DUE_high_err = 1.96*sqrt(datNormalized$FIT_DUE_high/datNormalized$number_DUE_high)

  datNormalized$FIT_DUE_thermals_err = 1.96*sqrt(datNormalized$FIT_DUE_thermals/datNormalized$number_DU
  datNormalized$FIT_SDC_thermals_err = 1.96*sqrt(datNormalized$FIT_SDC_thermals/datNormalized$number_SD

  datNormalized$FIT_SDC_thermals_storm_err = 1.96*sqrt(datNormalized$FIT_SDC_thermals_storm/datNormalize
  datNormalized$FIT_DUE_thermals_storm_err = 1.96*sqrt(datNormalized$FIT_DUE_thermals_storm/datNormalize
```

Plots

Cross Sections

Function to plot Cross Sections

Melt the data frame to plot two columns for each code and plot them with error bars

```
library(reshape2)

cs_sdc_plot_errbars <- function(device){
  datPlot = datNormalized[datNormalized$Device == device, ]

  dfm1 = melt(datPlot[, c('Code', 'cross_section_SDC_high_err', 'cross_section_SDC_high')], id.vars = c
  colnames(dfm1) = c("Code", "Error", "Neutron_Type", "Cross_Section")
  dfm1$Neutron_Type = "High Energy"
  dfm2 = melt(datPlot[, c('Code', 'cross_section_SDC_thermals_err', 'cross_section_SDC_thermals')], id.
  colnames(dfm2) = c("Code", "Error", "Neutron_Type", "Cross_Section")
  dfm2$Neutron_Type = "Thermals"
  dfm = rbind(dfm1, dfm2)

  plot = ggplot(dfm, aes(x=Code, weight=Cross_Section, ymin=Cross_Section-Error, ymax=Cross_Section+Err
  geom_bar      (position=position_dodge(), aes(y=Cross_Section), stat="identity") +
  geom_errorbar (position=position_dodge(width=0.9), width=0.2, colour="black") +
  labs(x = "", y = "Cross Section SDC [a.u.]", fill="") +
  theme(legend.position="bottom",axis.text.x=element_text(size=14) , axis.text.y=element_text(size=14)
  filename = paste("./plots/cs_SDC_", device, ".pdf", sep="")
  ggsave(filename, plot)
  plot
}

cs_due_plot_errbars <- function(device){
  datPlot = datNormalized[datNormalized$Device == device, ]

  dfm1 = melt(datPlot[, c('Code', 'cross_section_DUE_high_err', 'cross_section_DUE_high')], id.vars = c
  colnames(dfm1) = c("Code", "Error", "Neutron_Type", "Cross_Section")
  dfm1$Neutron_Type = "High Energy"
  dfm2 = melt(datPlot[, c('Code', 'cross_section_DUE_thermals_err', 'cross_section_DUE_thermals')], id.
  colnames(dfm2) = c("Code", "Error", "Neutron_Type", "Cross_Section")
  dfm2$Neutron_Type = "Thermals"
  dfm = rbind(dfm1, dfm2)
  plot = ggplot(dfm, aes(x=Code, weight=Cross_Section, ymin=Cross_Section-Error, ymax=Cross_Section+Erro
    geom_bar      (position=position_dodge(), aes(y=Cross_Section), stat="identity") +
    geom_errorbar (position=position_dodge(width=0.9), width=0.2, colour="black") +
    #labs(title = plot_title, x = "", y = "Cross Section DUE [a.u.]", fill="") +
    labs(x = "", y = "Cross Section DUE [a.u.]", fill="") +
    theme(legend.position="bottom",axis.text.x=element_text(size=14) , axis.text.y=element_text(size=14
  #theme(plot.title = element_text(hjust = 0.5), legend.position="bottom",axis.text.x=element_text(size
  filename = paste("./plots/cs_SDC_", device, ".pdf", sep="")
  ggsave(filename, plot)
  plot
}
```

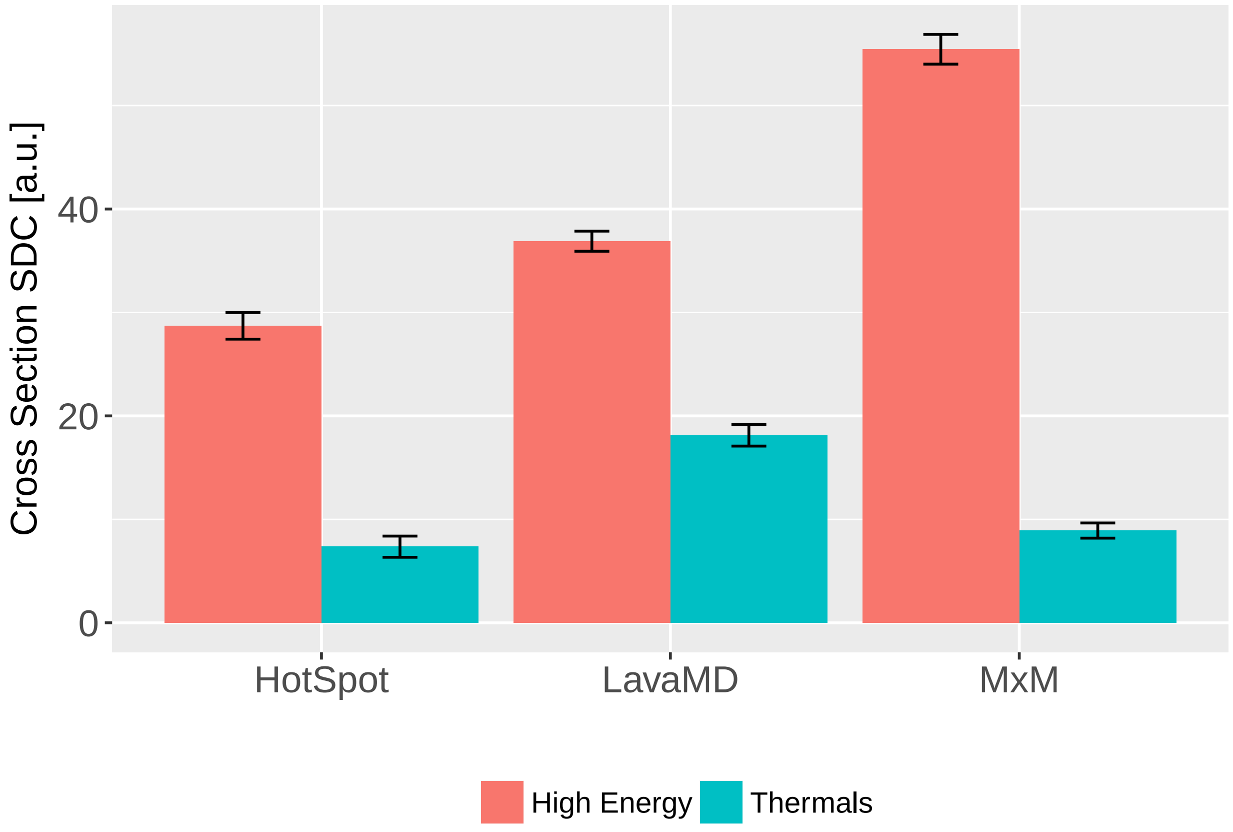
Plot Cross Sections

Titan X

SDC

```
cs_sdc_plot_errbars("TitanX")
```

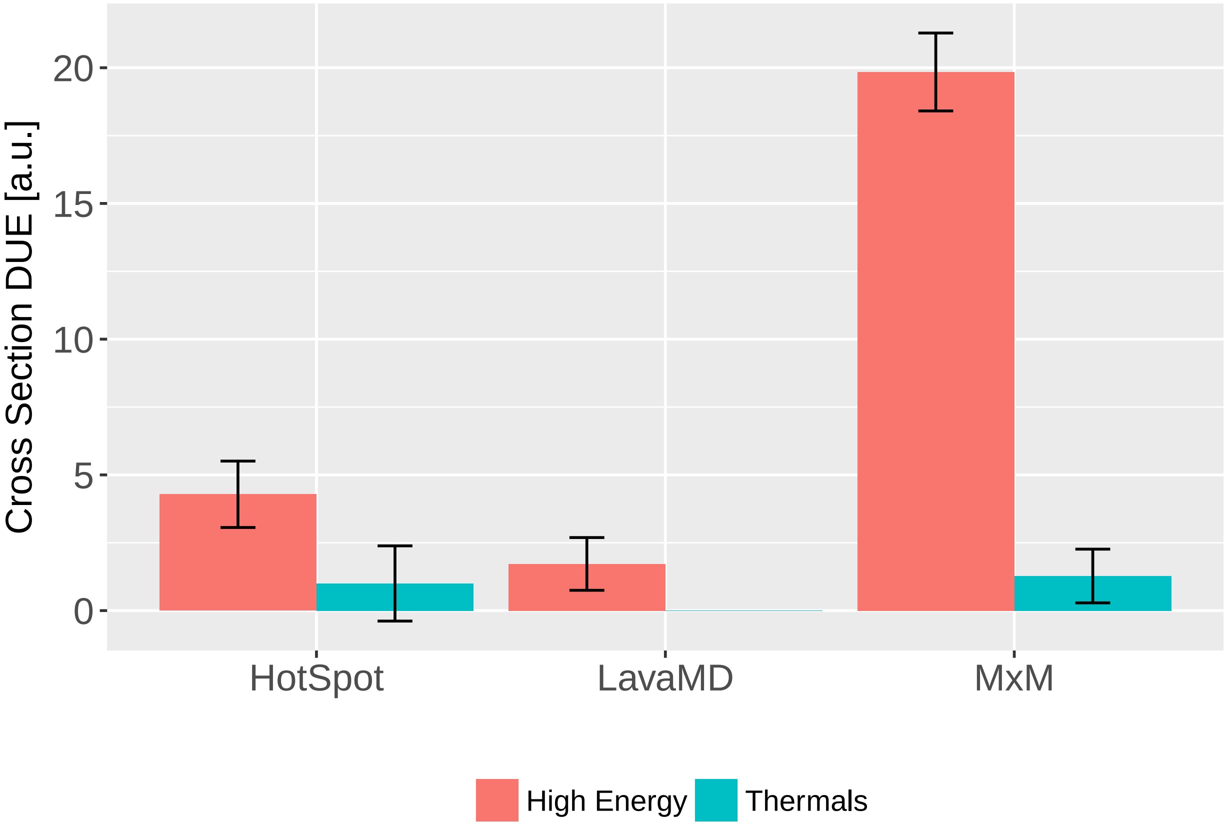
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_plot_errbars("TitanX")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 1 rows containing missing values (geom_errorbar).

## Warning: Removed 1 rows containing missing values (geom_errorbar).
```

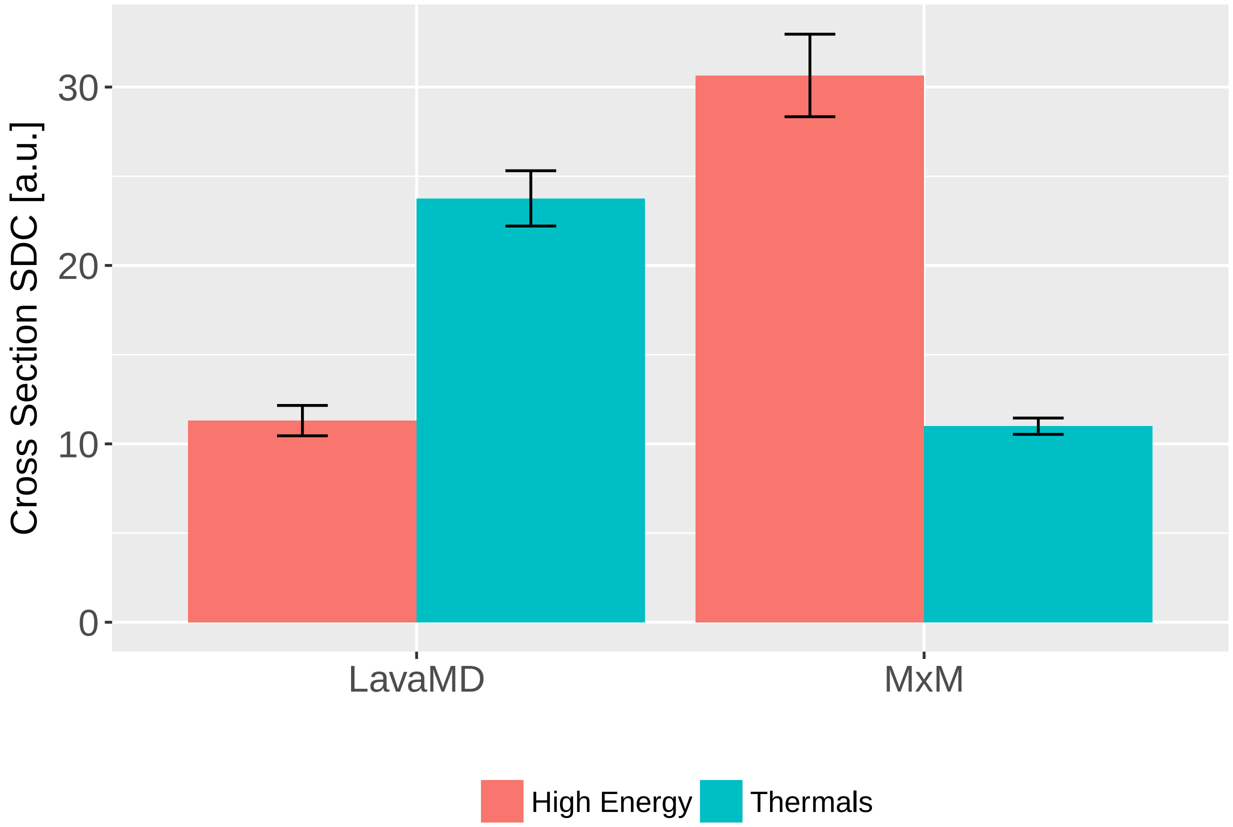


Titan V

SDC

```
cs_sdc_plot_errbars("TitanV")
```

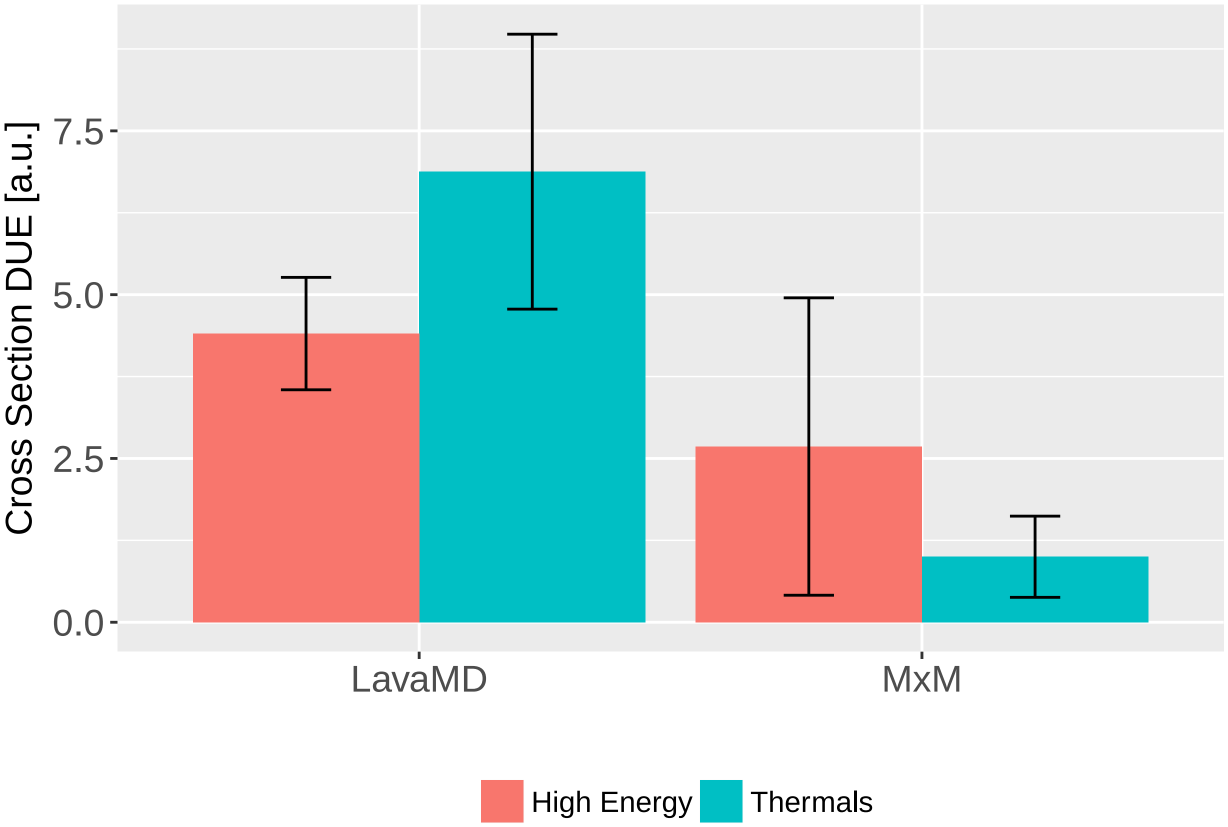
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_plot_errbars("TitanV")
```

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

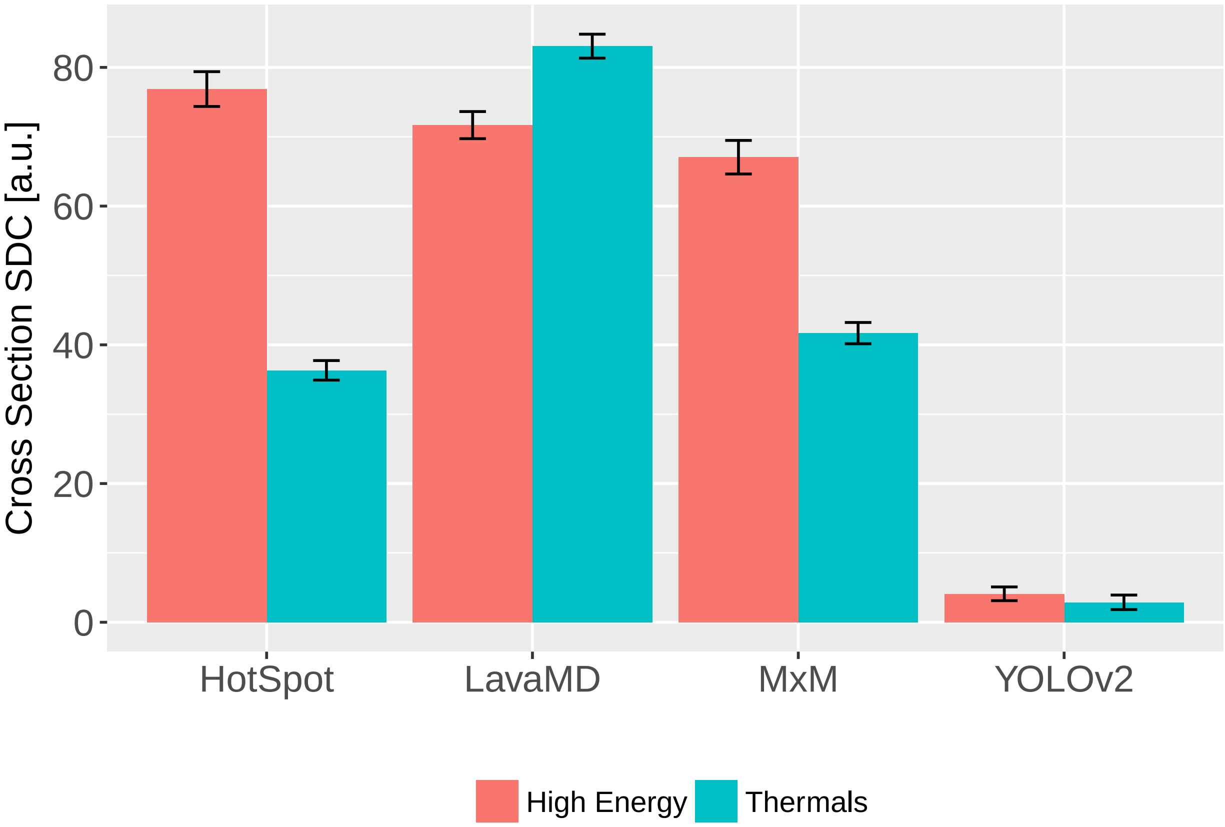


K20

SDC

```
cs_sdc_plot_errbars("K20")
```

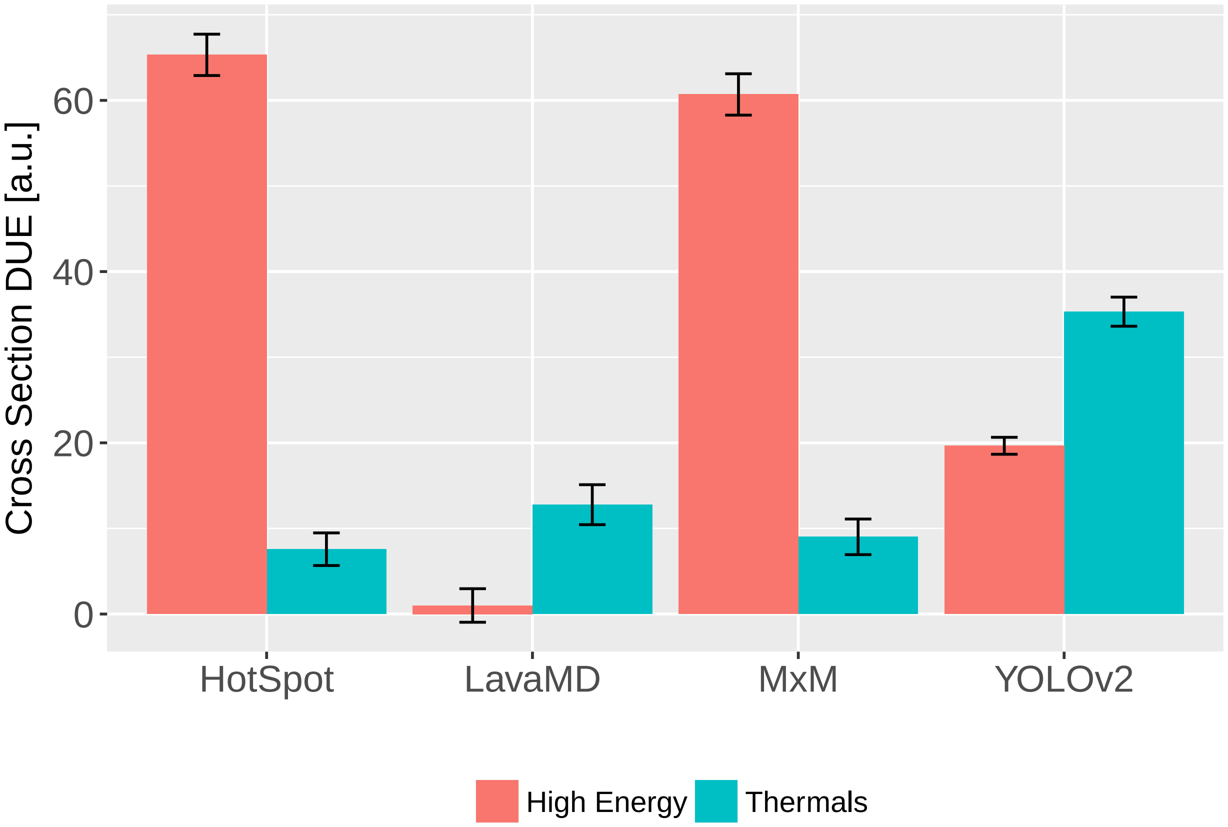
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_plot_errbars("K20")
```

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

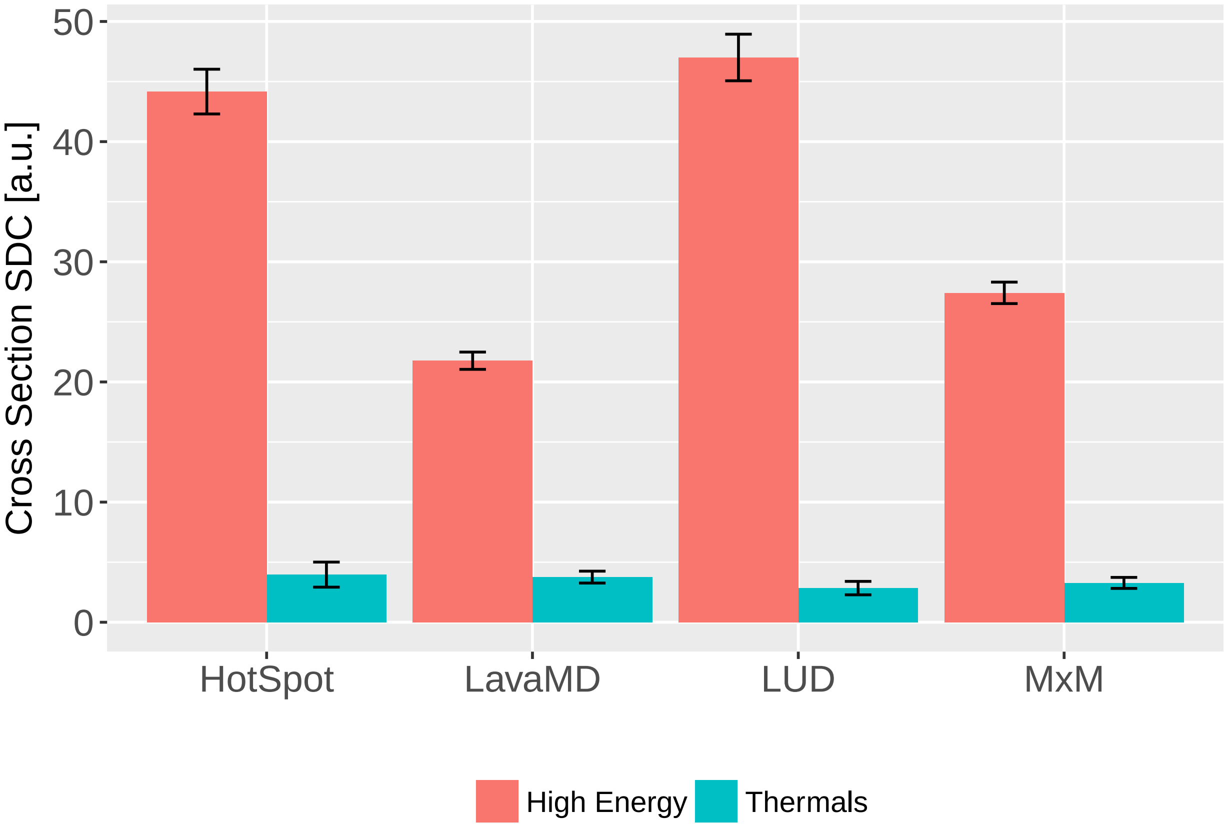


Xeon Phi

SDC

```
cs_sdc_plot_errbars("XeonPhi")
```

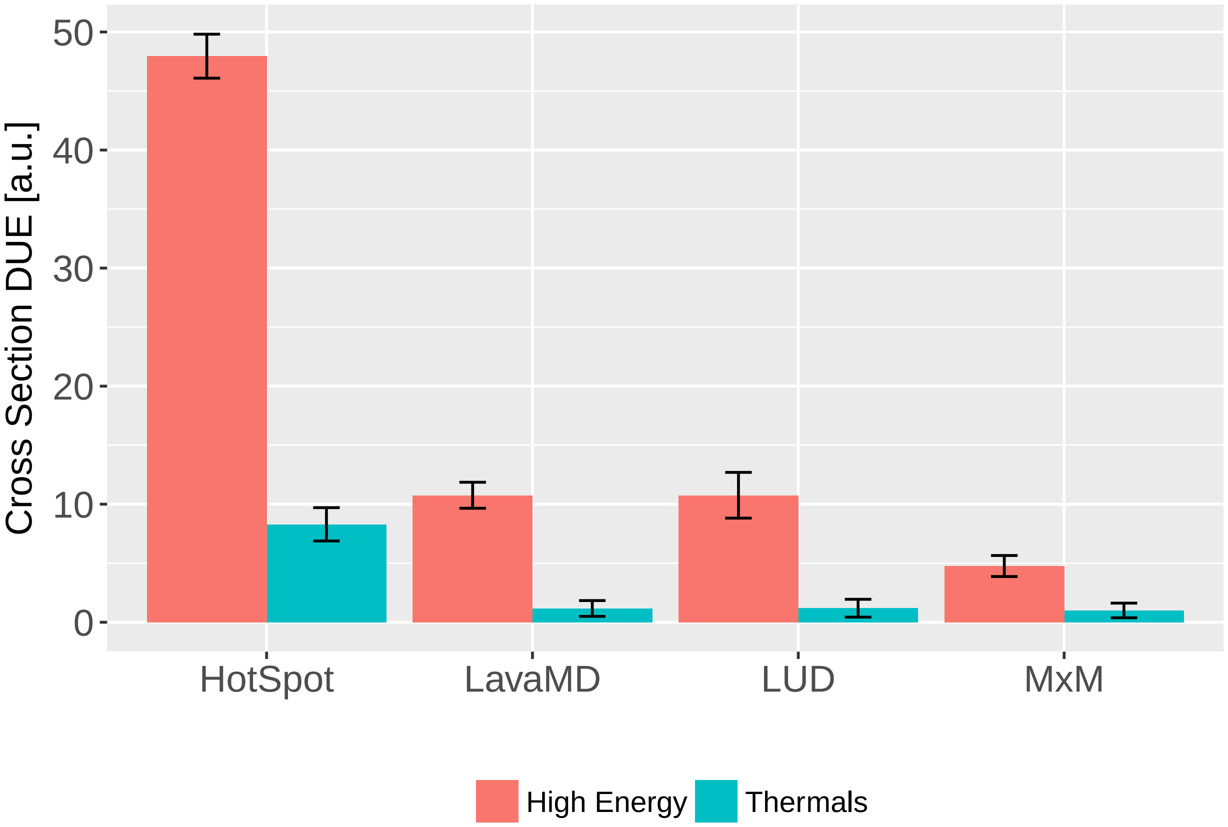
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_plot_errbars("XeonPhi")
```

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

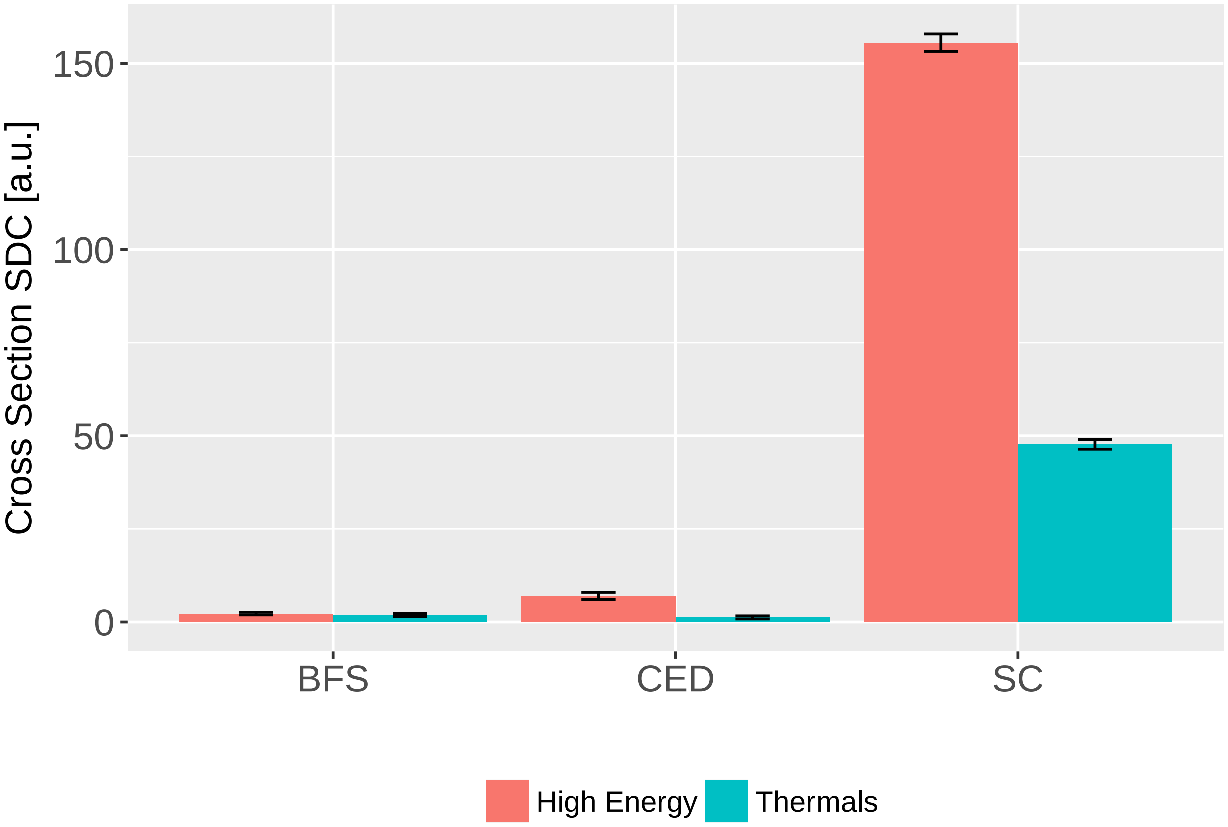


CPU

SDC

```
cs_sdc_plot_errbars("CPU")
```

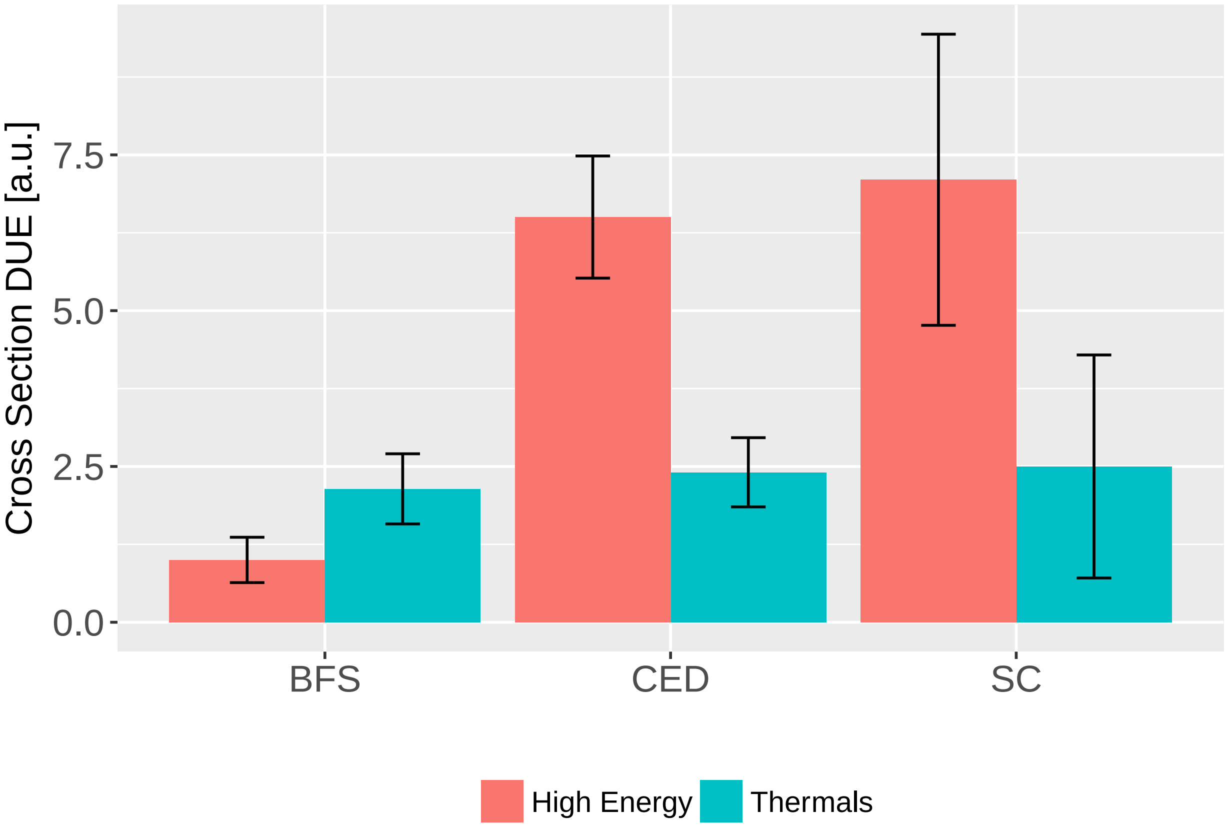
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_plot_errbars("CPU")
```

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

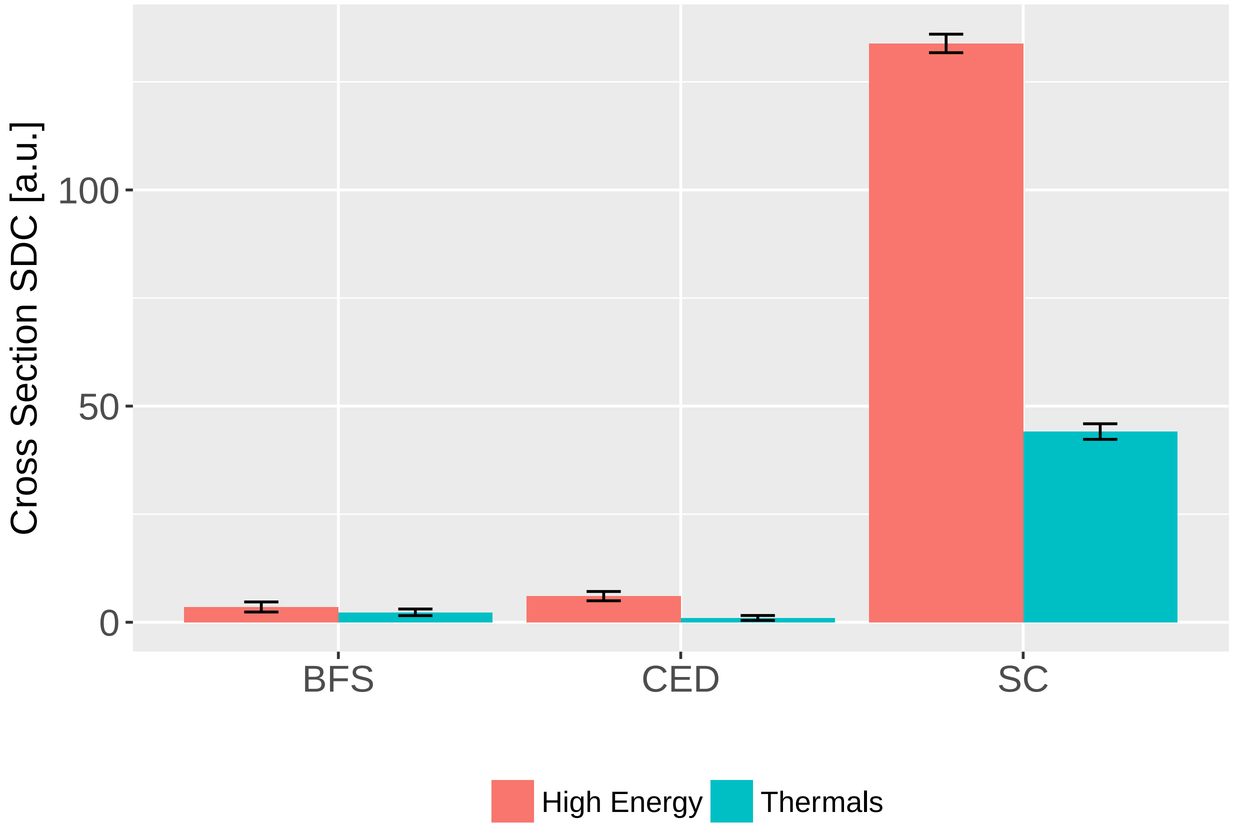


GPU

SDC

```
cs_sdc_plot_errbars("GPUembedded")
```

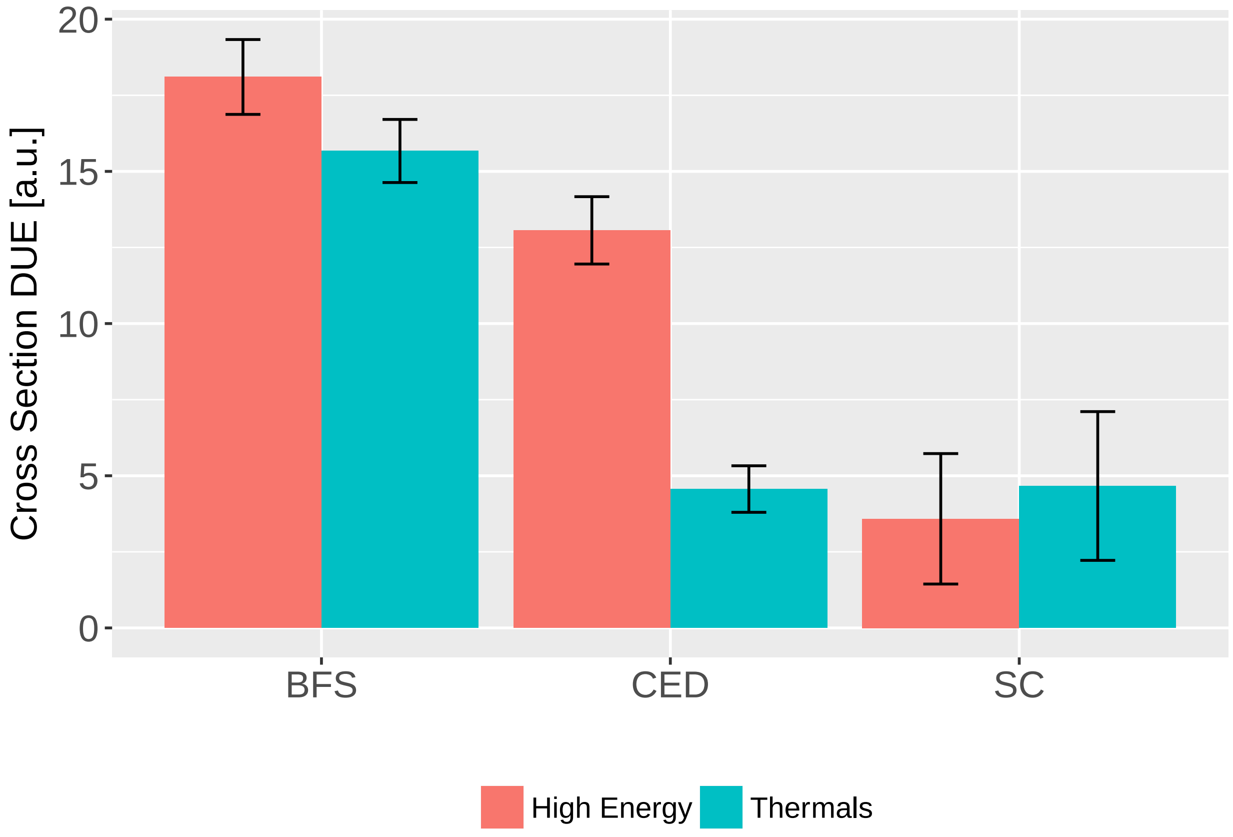
```
## Saving 6.5 x 4.5 in image
```



DUE

```r
cs_due_plot_errbars("GPUembedded")
```

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

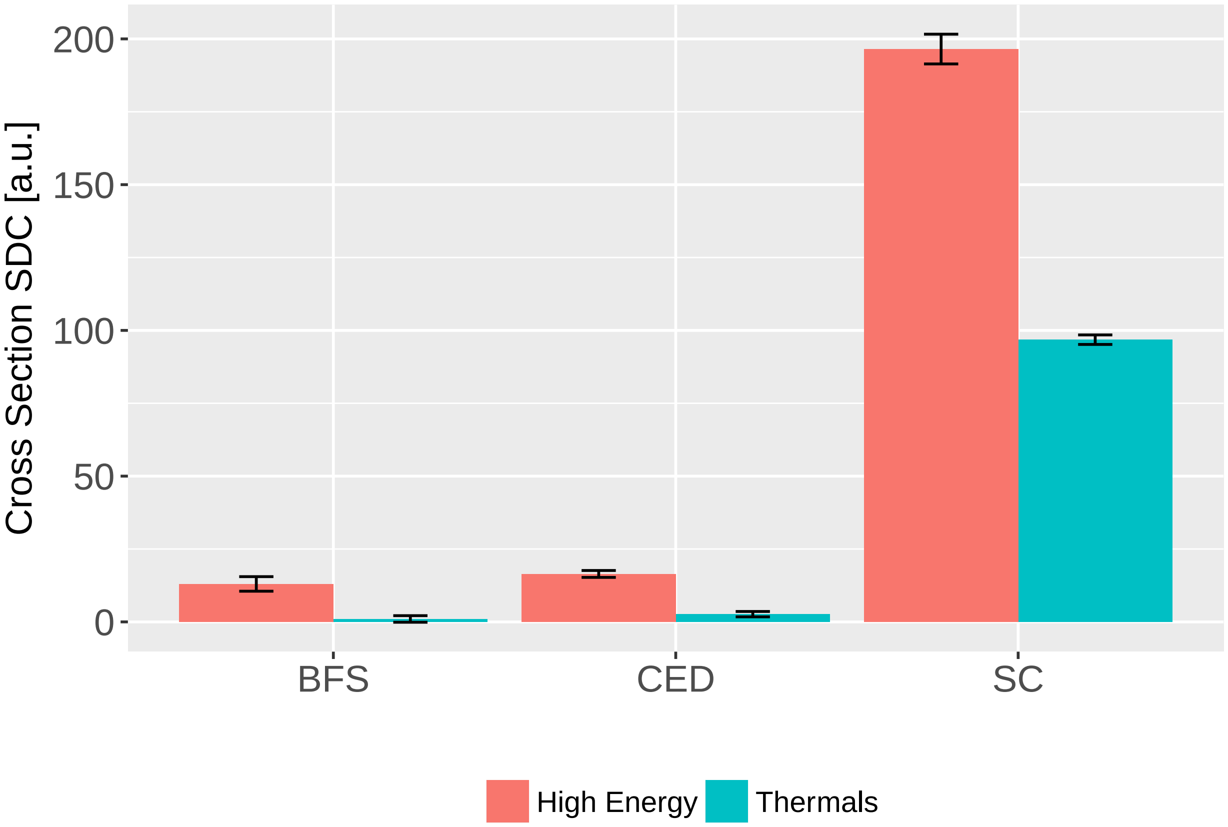


CPU+GPU

SDC

```
cs_sdc_plot_errbars("CPU+GPU")
```

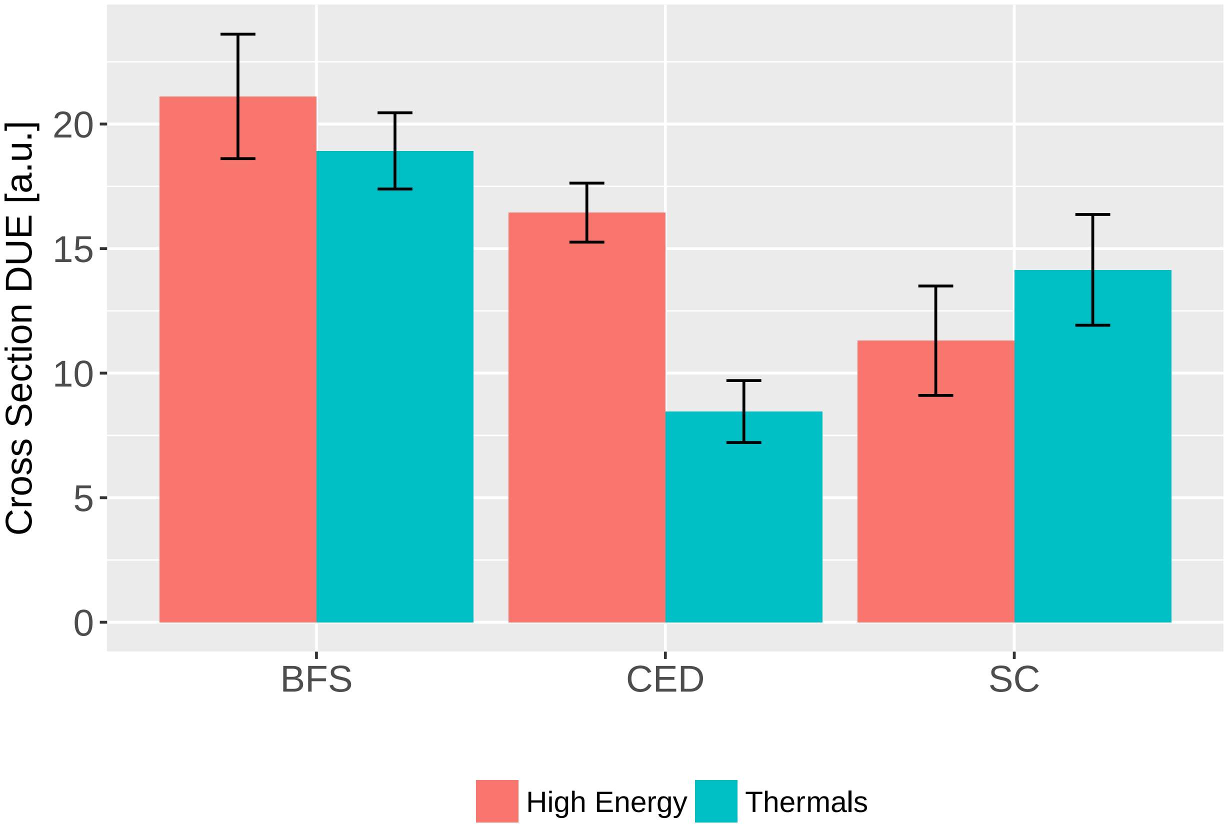
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_plot_errbars("CPU+GPU")
```

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



Cross Section Ratio

The idea is to show the difference between the cross sections of high energy and thermals in a more “easy to compare” way. Thus, we will divide the larger one by the smaller one to show the magnitude of the difference.

First we compute the ration and save into a new column the ratio and another one which neutron type (High Energy or Thermals) is the larger one.

```
datNormalized$ratioSDC = ifelse(datNormalized$cross_section_SDC_high >= datNormalized$cross_section_SDC_
datNormalized$moreSensitiveSDC = ifelse(datNormalized$cross_section_SDC_high >= datNormalized$cross_sec

datNormalized$ratioDUE = ifelse(datNormalized$cross_section_DUE_high >= datNormalized$cross_section_DUE_
datNormalized$moreSensitiveDUE = ifelse(datNormalized$cross_section_DUE_high >= datNormalized$cross_sec

datNormalized$ratioDUE = ifelse(datNormalized$cross_section_DUE_high == 0 | datNormalized$cross_section_
```

Now the functions to plot the data

```
cs_sdc_ratio <- function(device){
  plot = ggplot(datNormalized[datNormalized$Device == device, ], aes(x=Code, y=ratioSDC, fill=moreSensi
    geom_bar(stat="identity") +
    labs(x = "", y = "Cross Section SDC Ratio", fill="") +
    theme(legend.position="bottom",axis.text.x=element_text(size=14) , axis.text.y=element_text(size=14)

  filename = paste("./plots/cs_ratio_SDC_", device, ".pdf", sep="")
  ggsave(filename, plot)
  plot
}
```

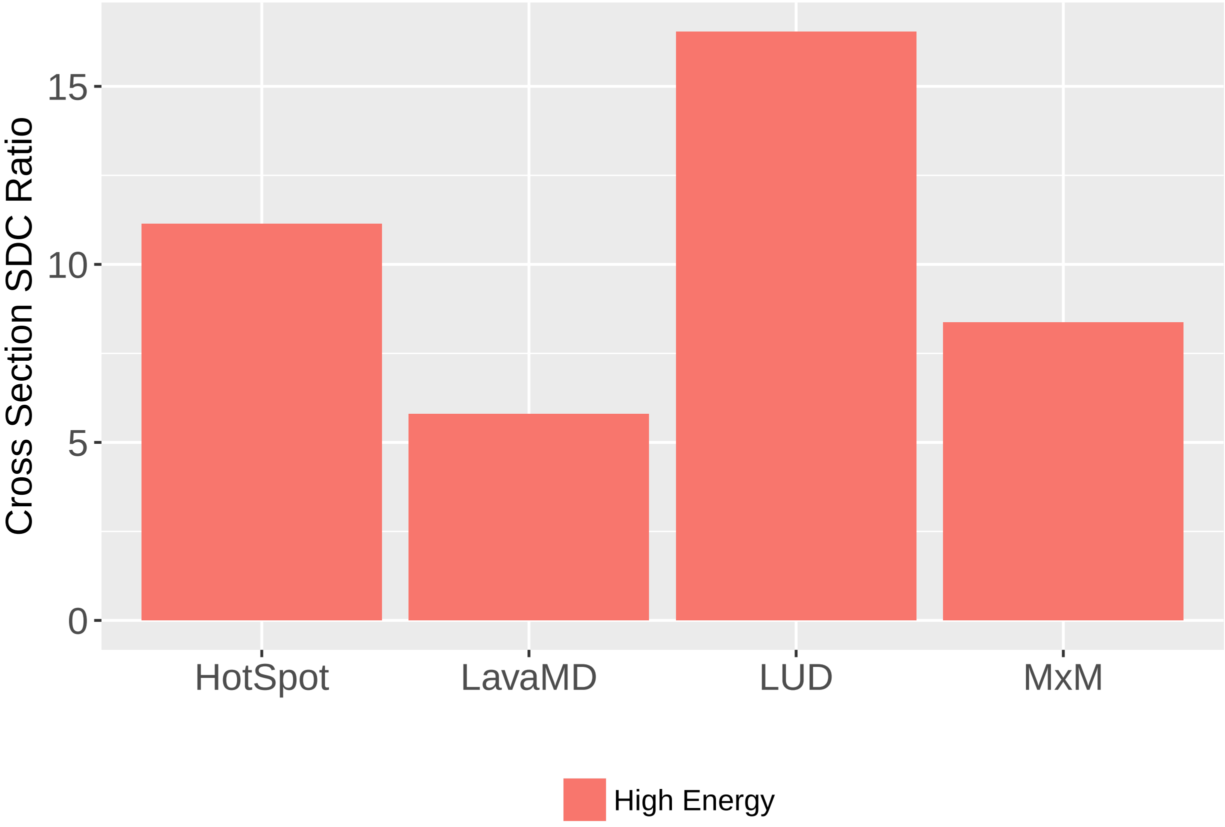
```
cs_due_ratio <- function(device){
  plot = ggplot(datNormalized[datNormalized$Device == device, ], aes(x=Code, y=ratioDUE, fill=moreSensi
    geom_bar(stat="identity") +
    labs(x = "", y = "Cross Section DUE Ratio", fill="") +
    theme(legend.position="bottom",axis.text.x=element_text(size=14) , axis.text.y=element_text(size=14

  filename = paste("./plots/cs_ratio_DUE_", device, ".pdf", sep="")
  ggsave(filename, plot)
  plot
}
```

Xeon Phi

SDC

```
cs_sdc_ratio("XeonPhi")
```

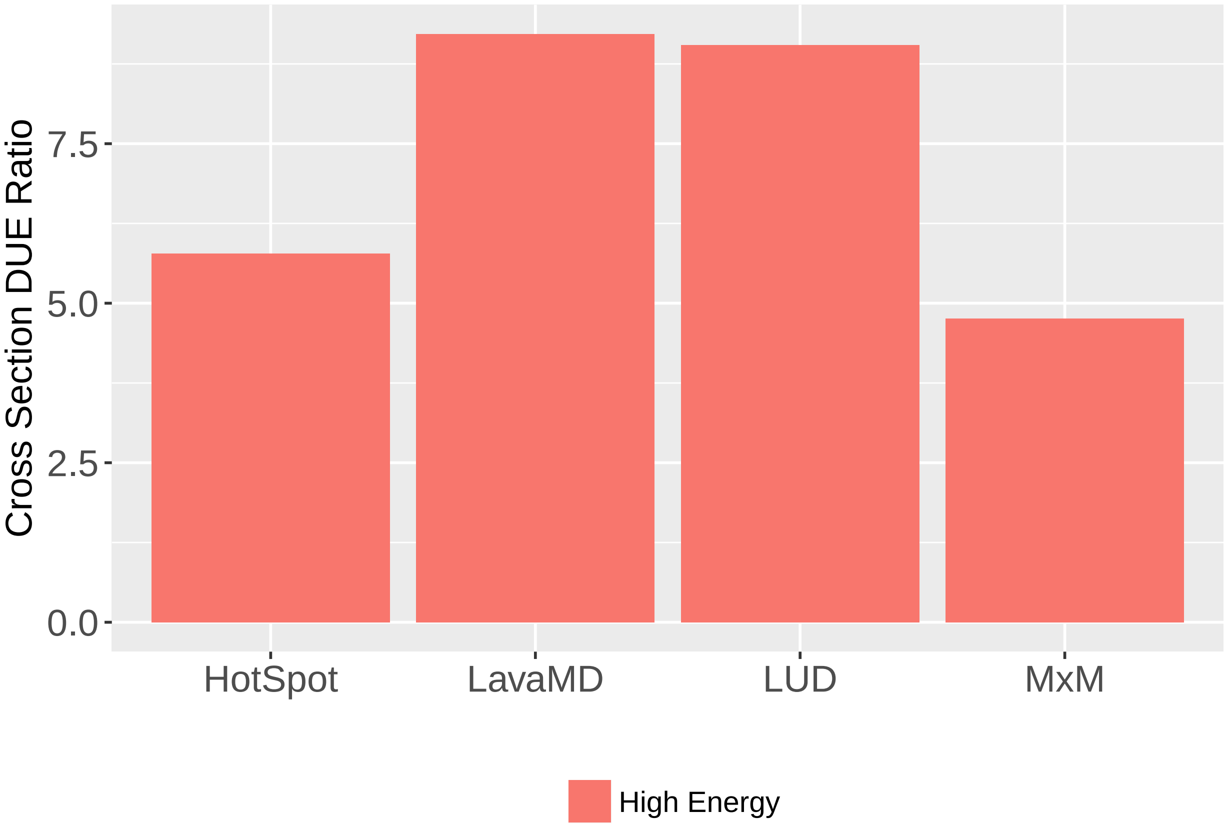
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_ratio("XeonPhi")
```

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

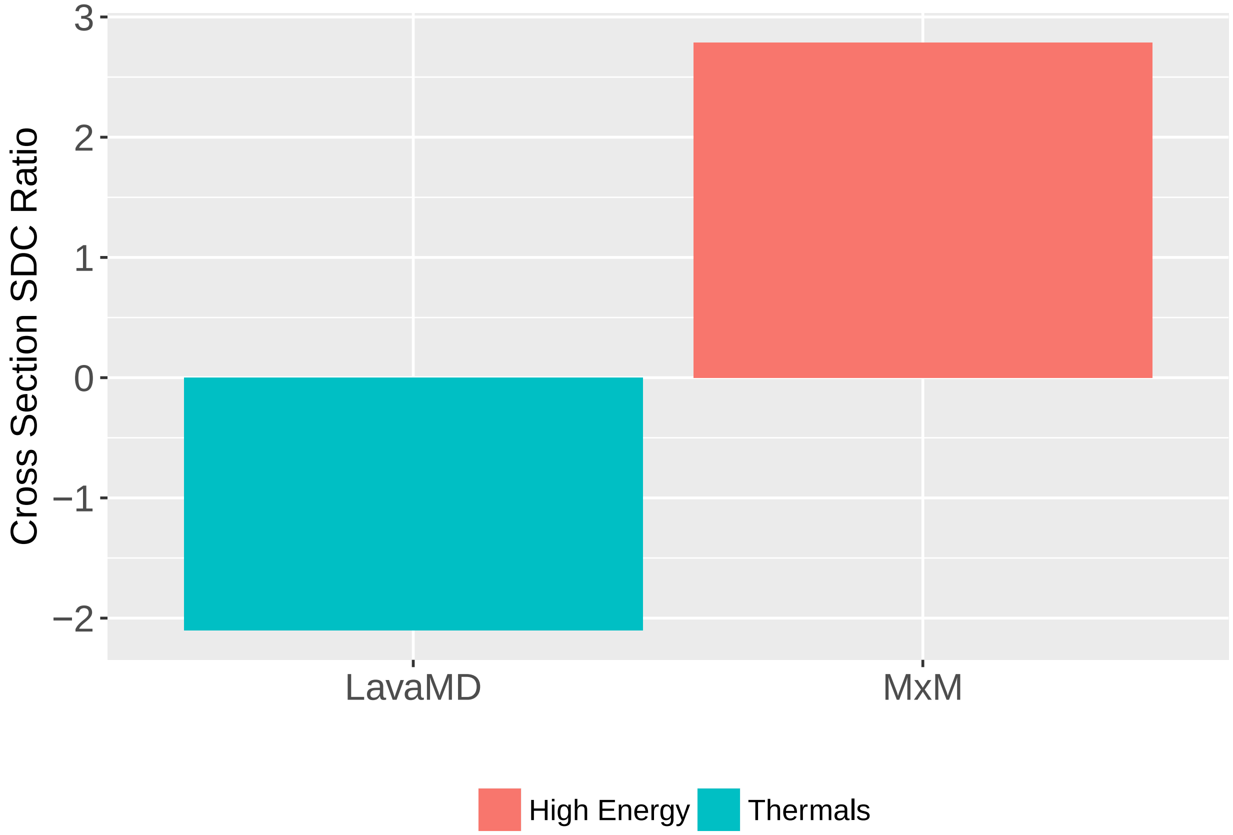


TitanV

SDC

```
cs_sdc_ratio("TitanV")
```

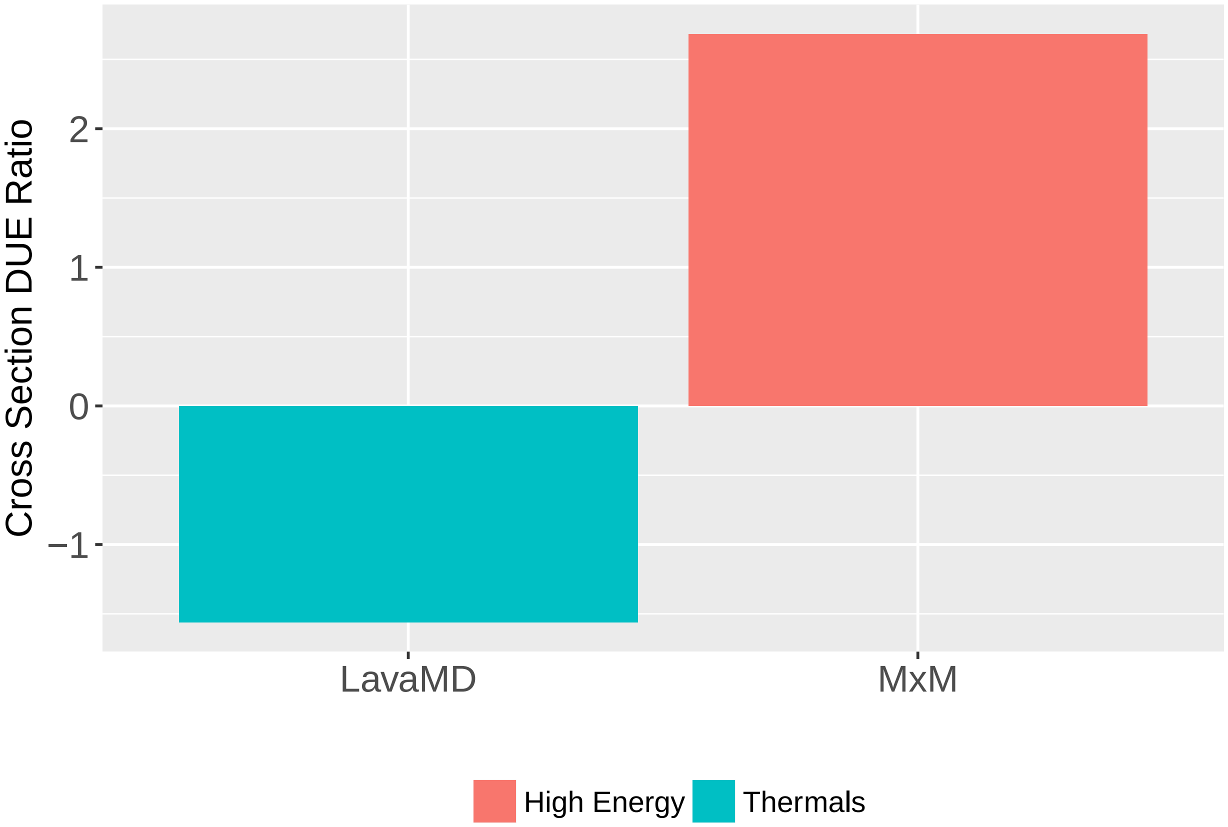
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_ratio("TitanV")
```

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

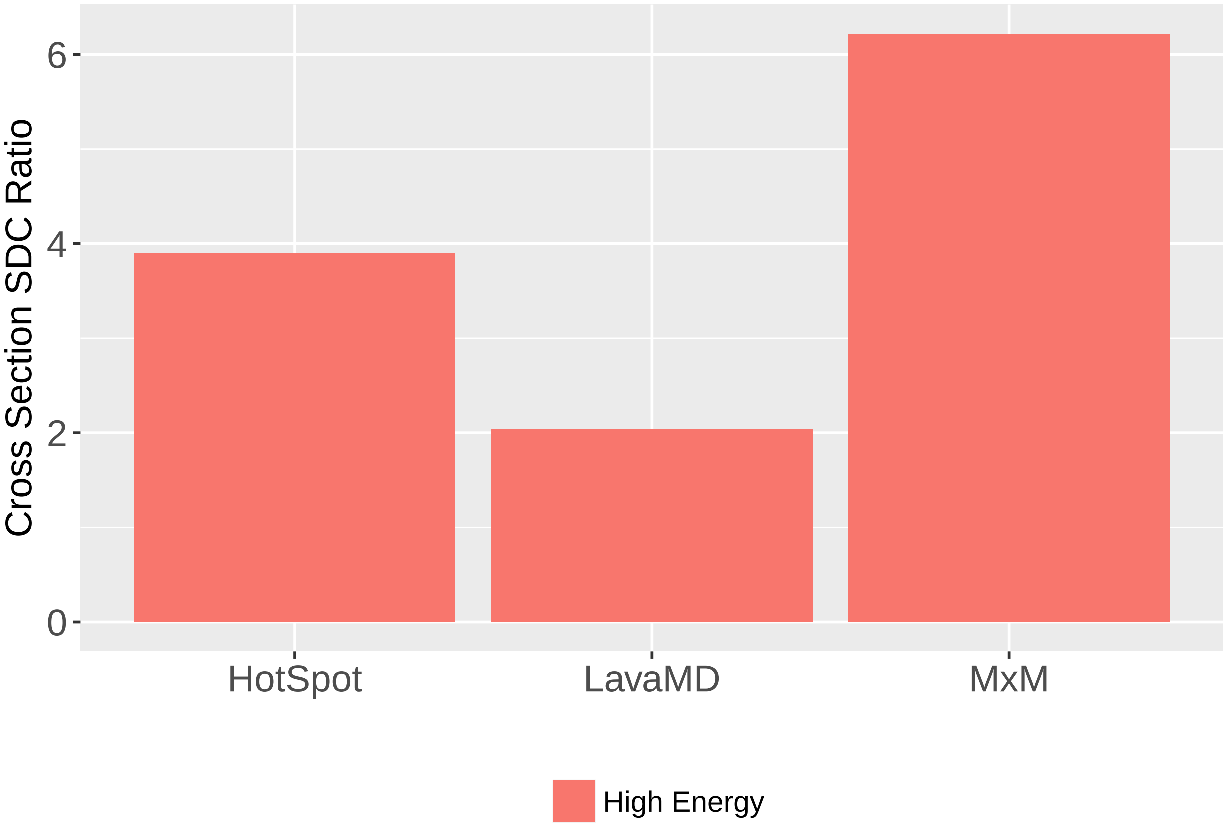


TitanX

SDC

```
cs_sdc_ratio("TitanX")
```

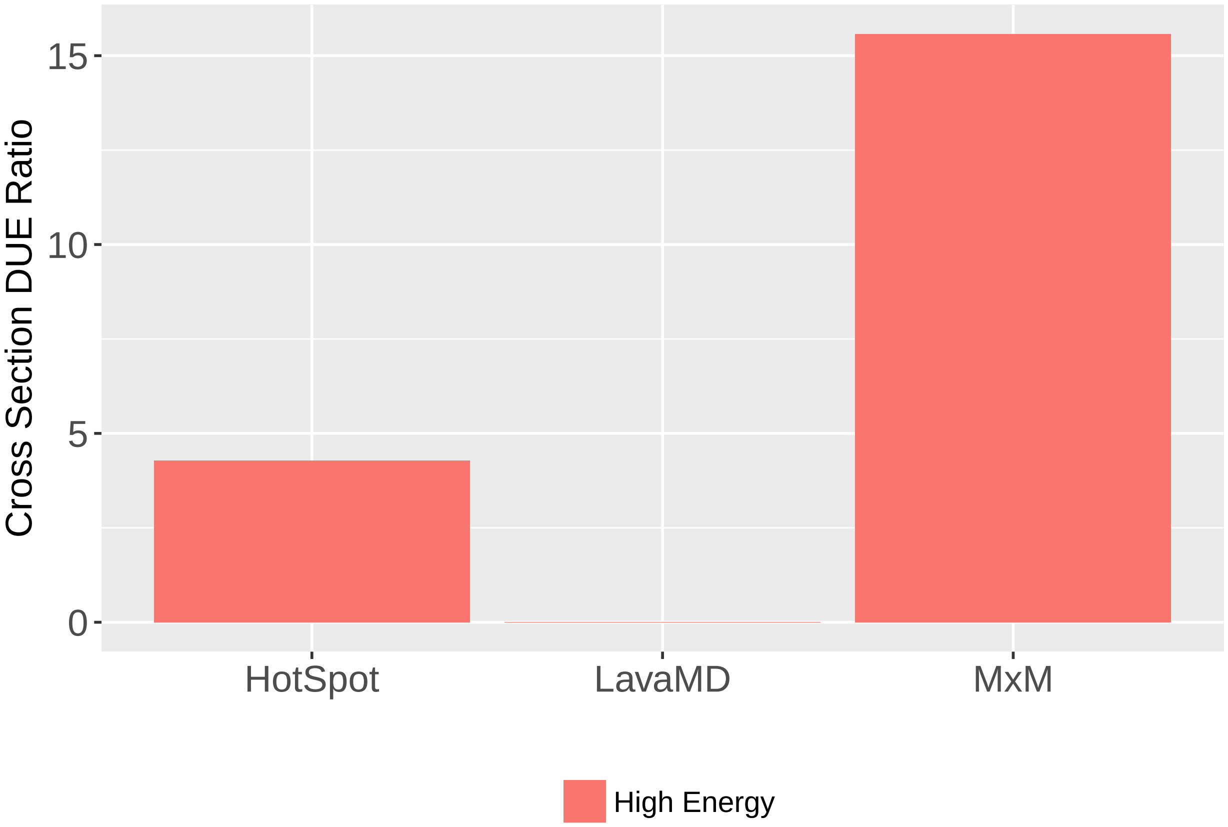
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_ratio("TitanX")
```

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

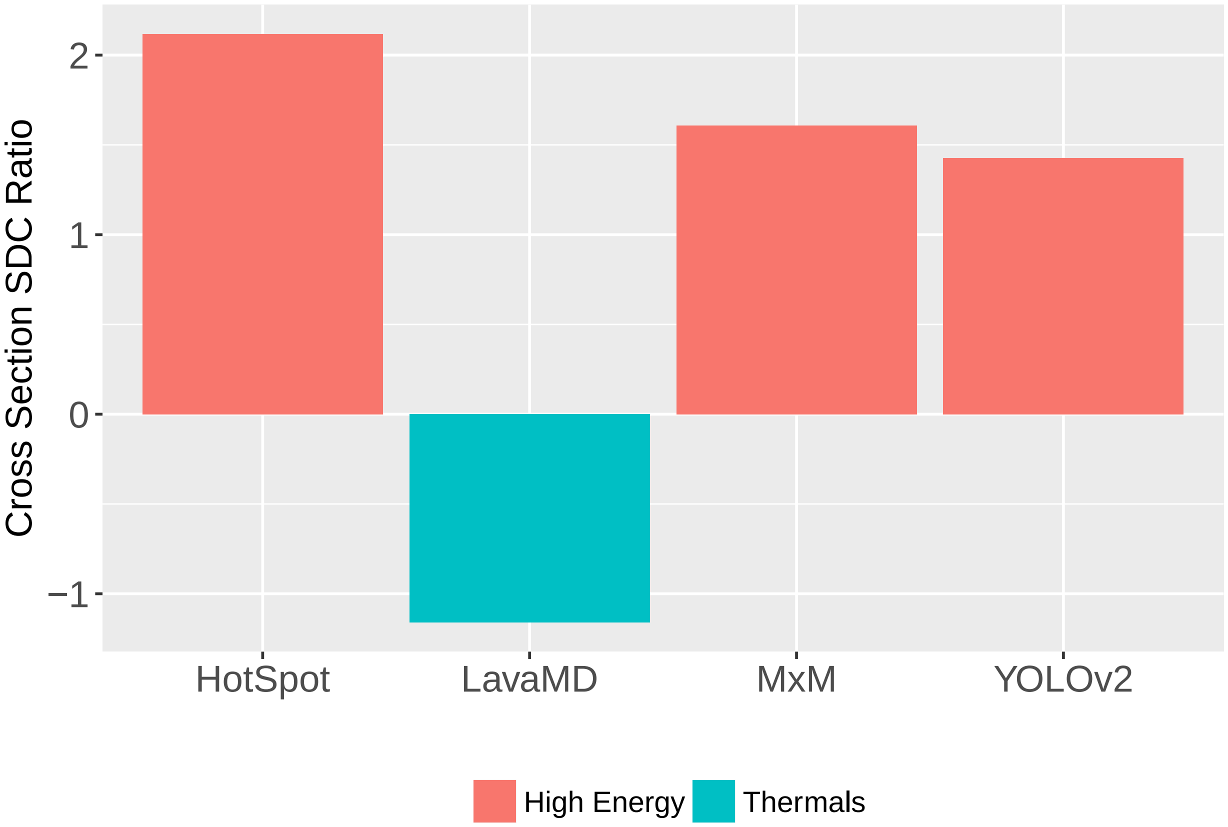


K20

SDC

```
cs_sdc_ratio("K20")
```

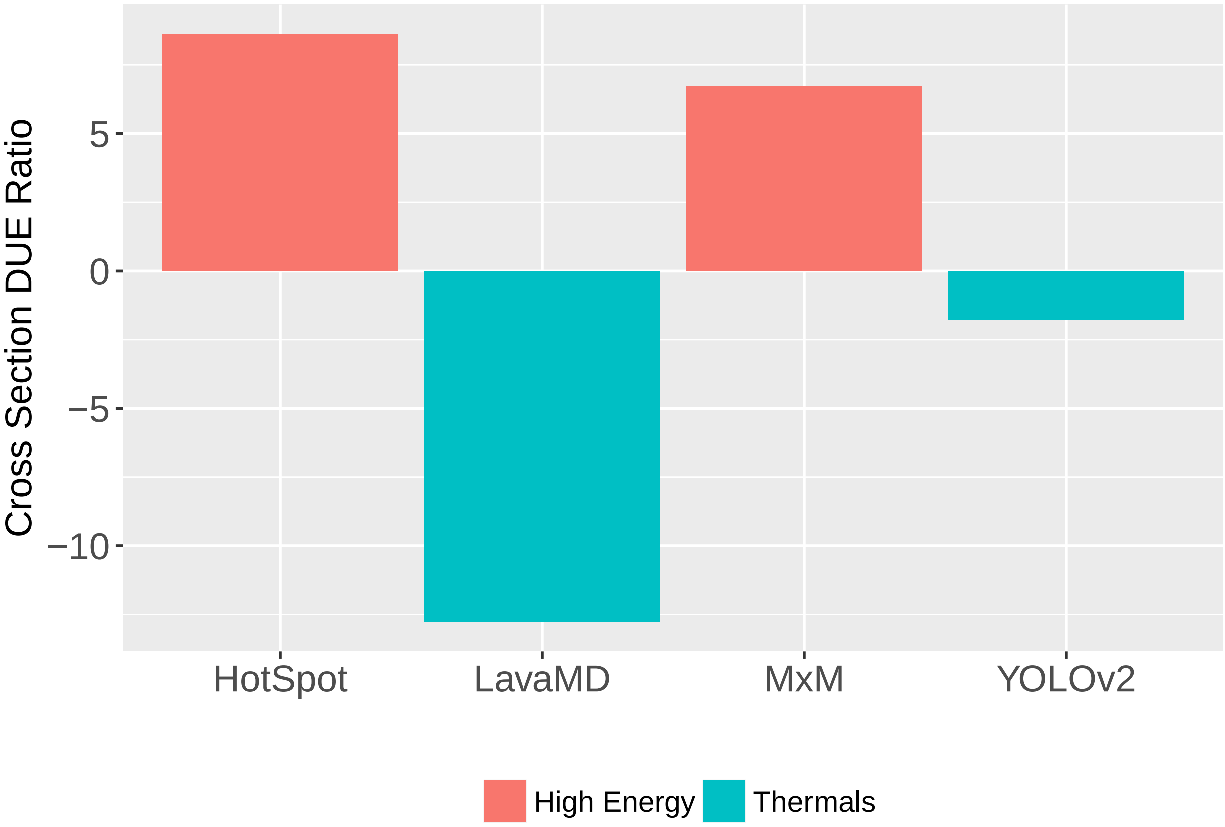
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_ratio("K20")
```

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

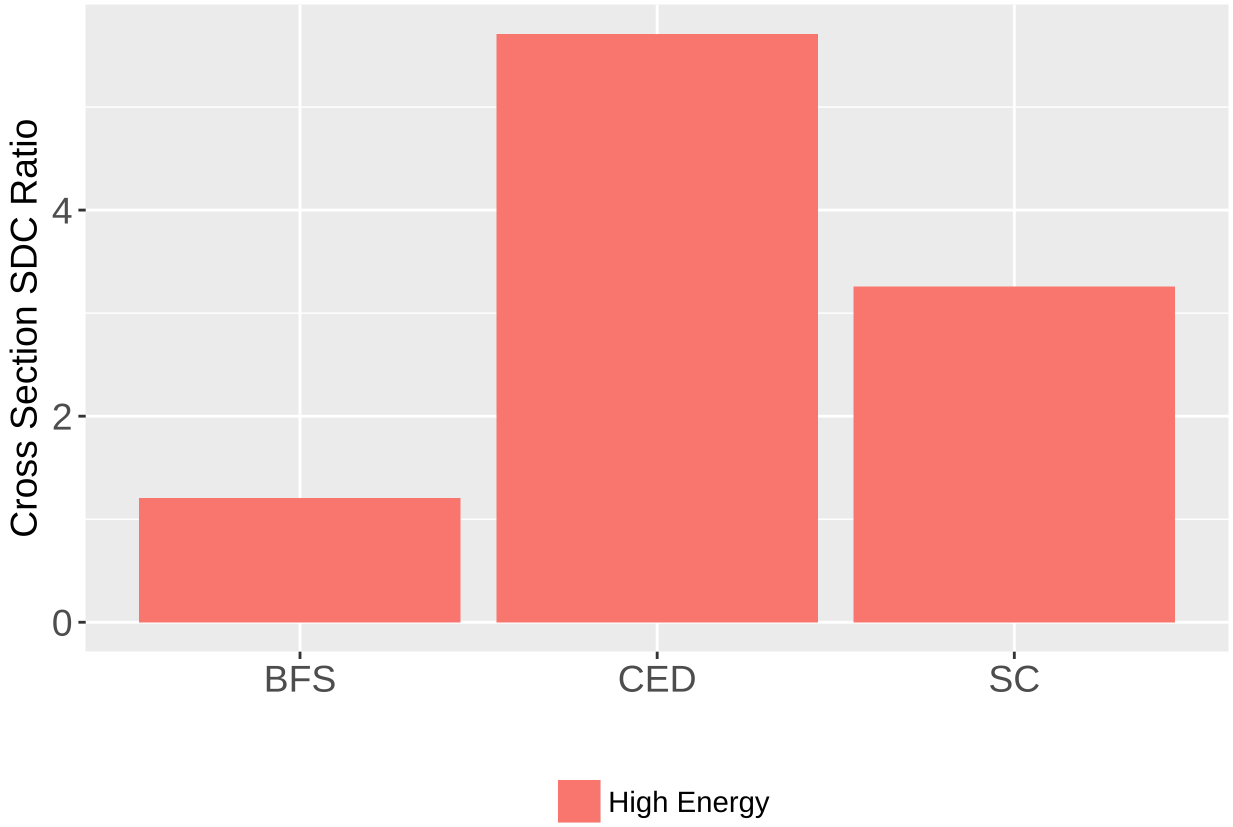


CPU

SDC

```
cs_sdc_ratio("CPU")
```

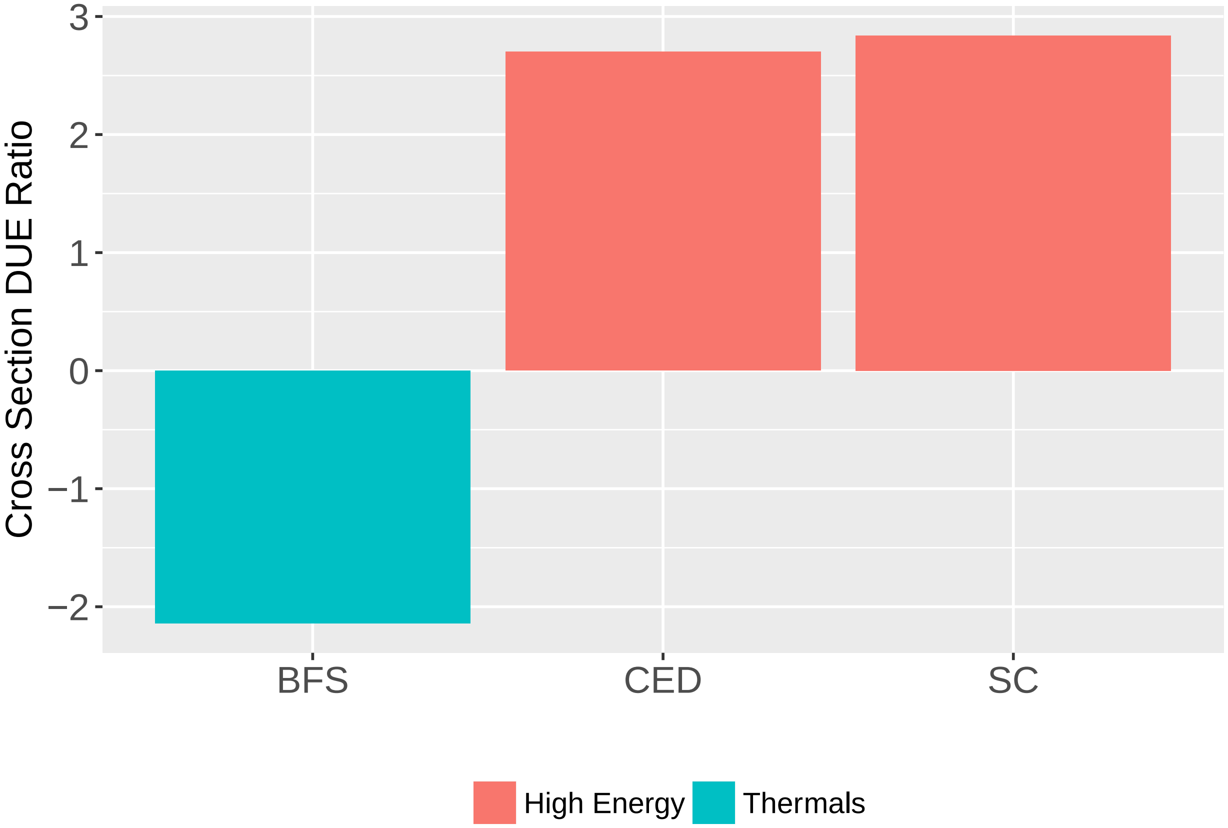
```
## Saving 6.5 x 4.5 in image
```



DUE

```
cs_due_ratio("CPU")
```

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

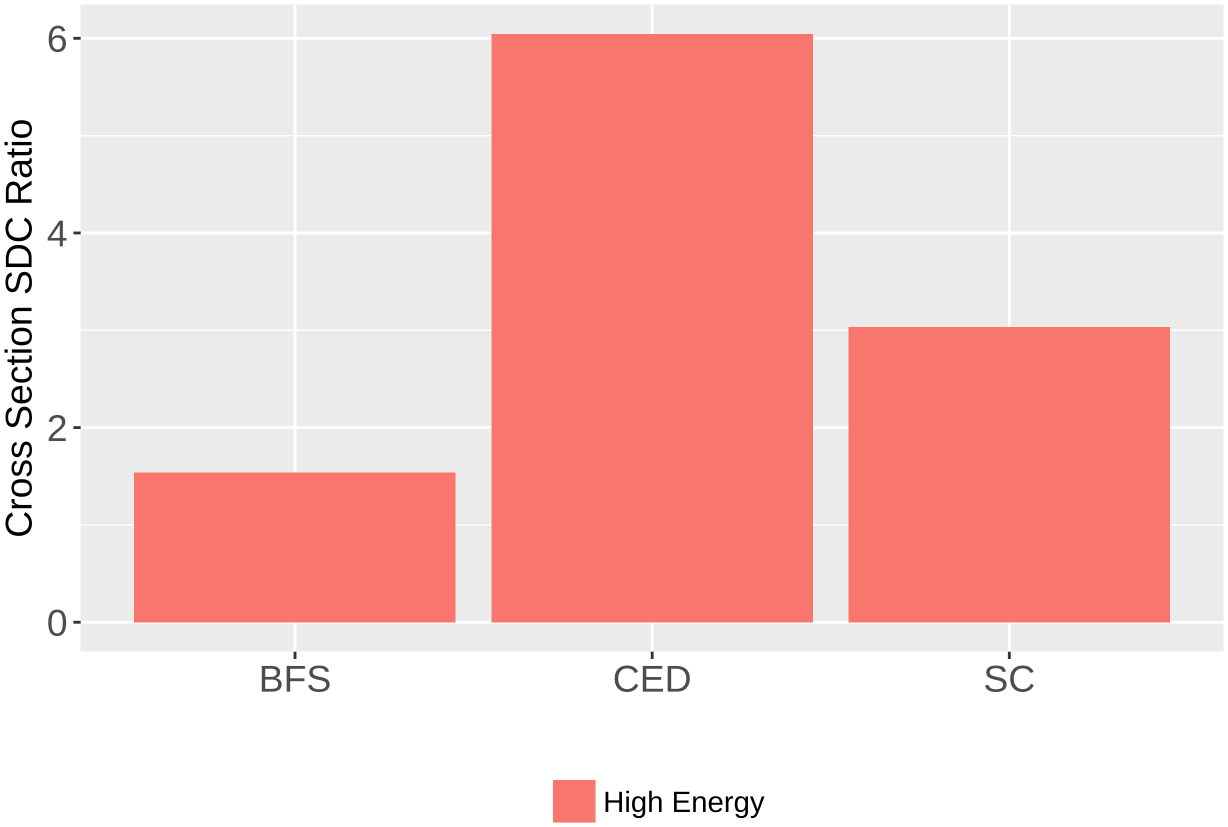


GPUembedded

SDC

```
cs_sdc_ratio("GPUembedded")
```

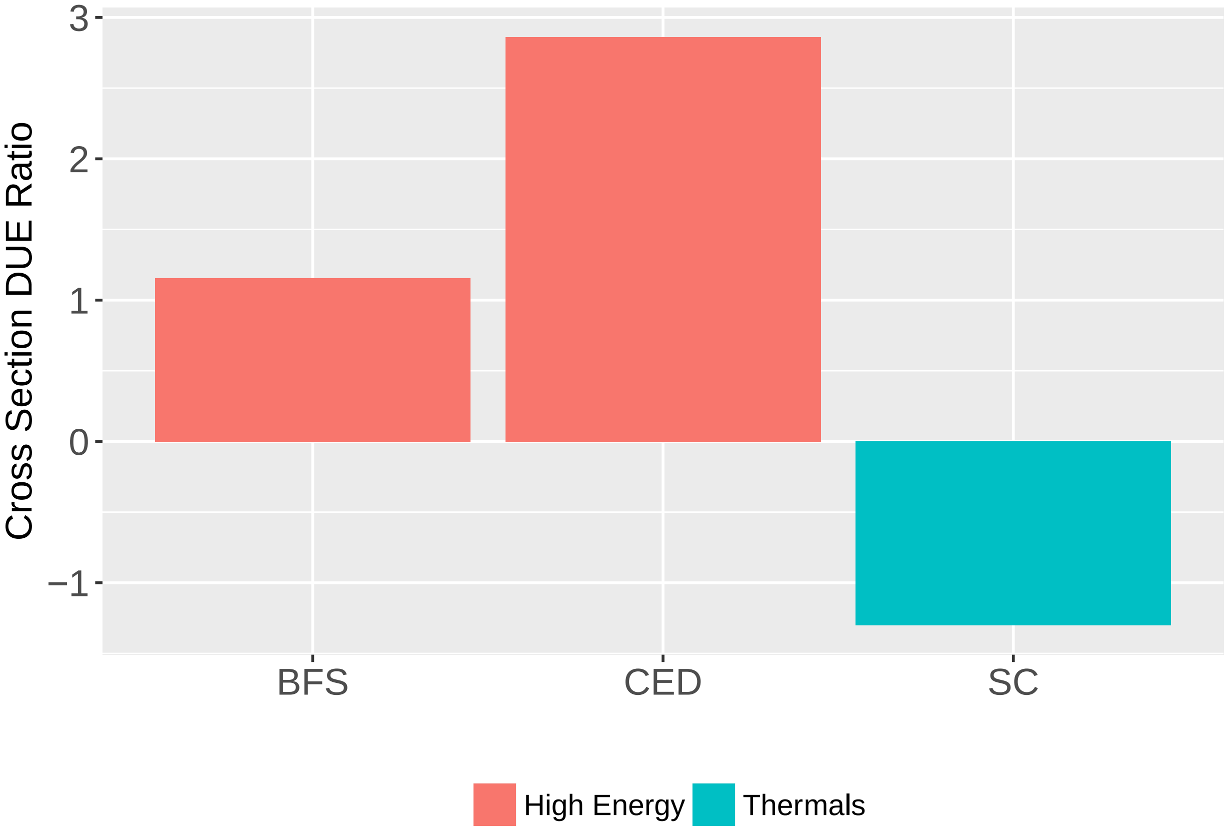
```
## Saving 6.5 x 4.5 in image
```



DUE

```r
cs_due_ratio("GPUembedded")
```

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

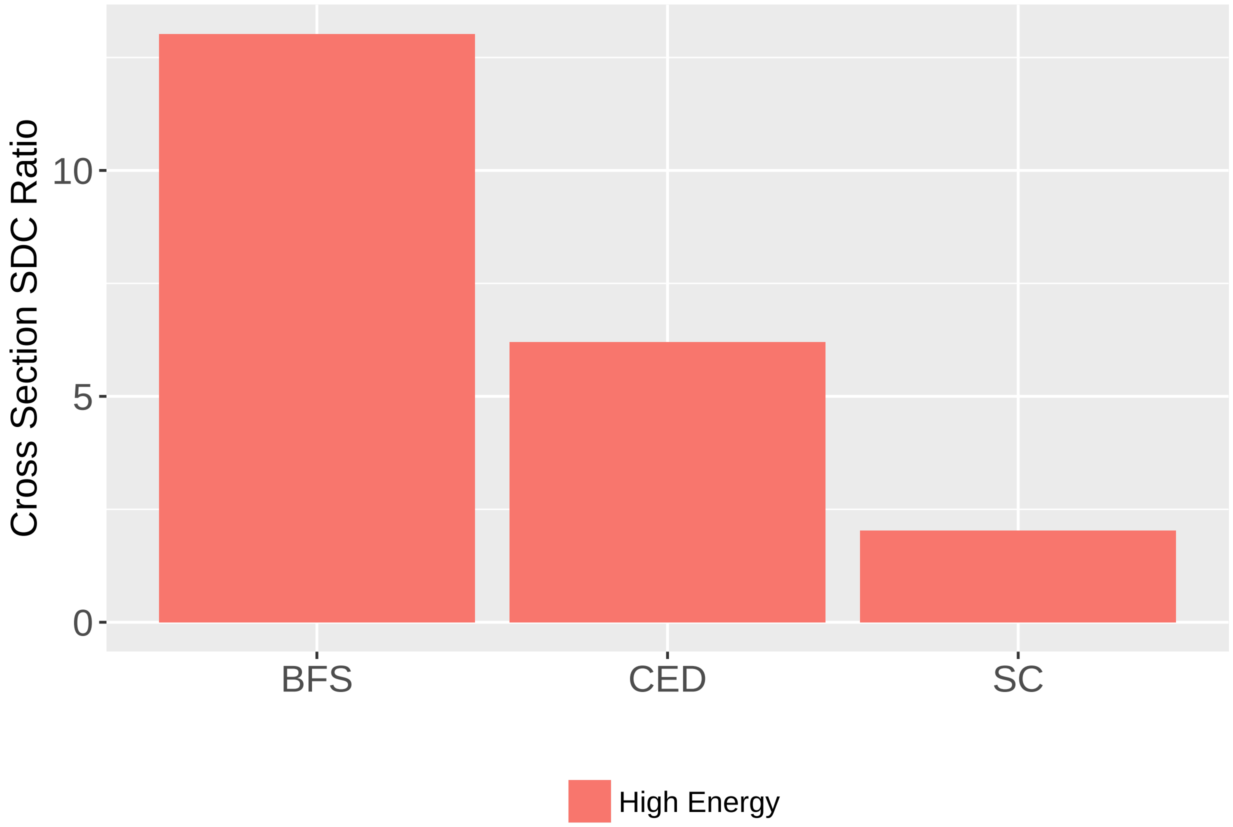


CPU+GPU

SDC

```
cs_sdc_ratio("CPU+GPU")
```

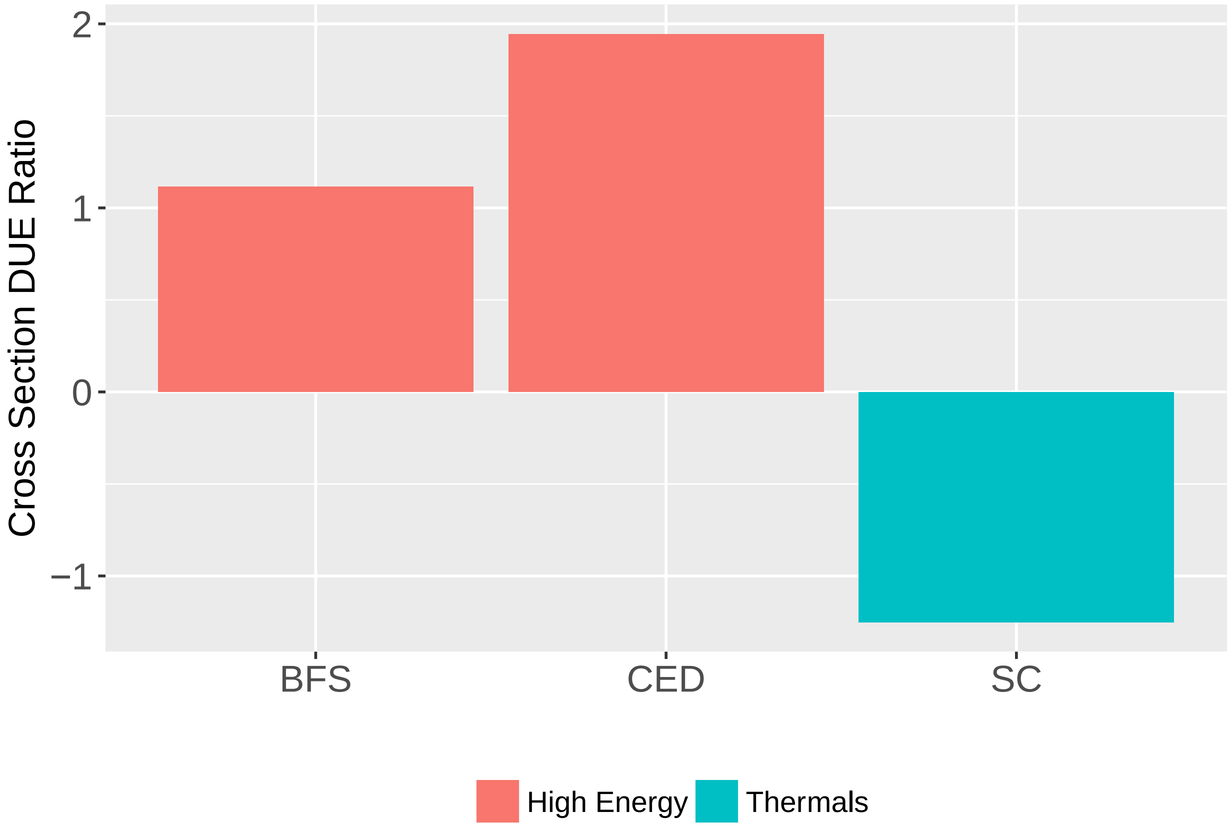
```
## Saving 6.5 x 4.5 in image
```



DUE

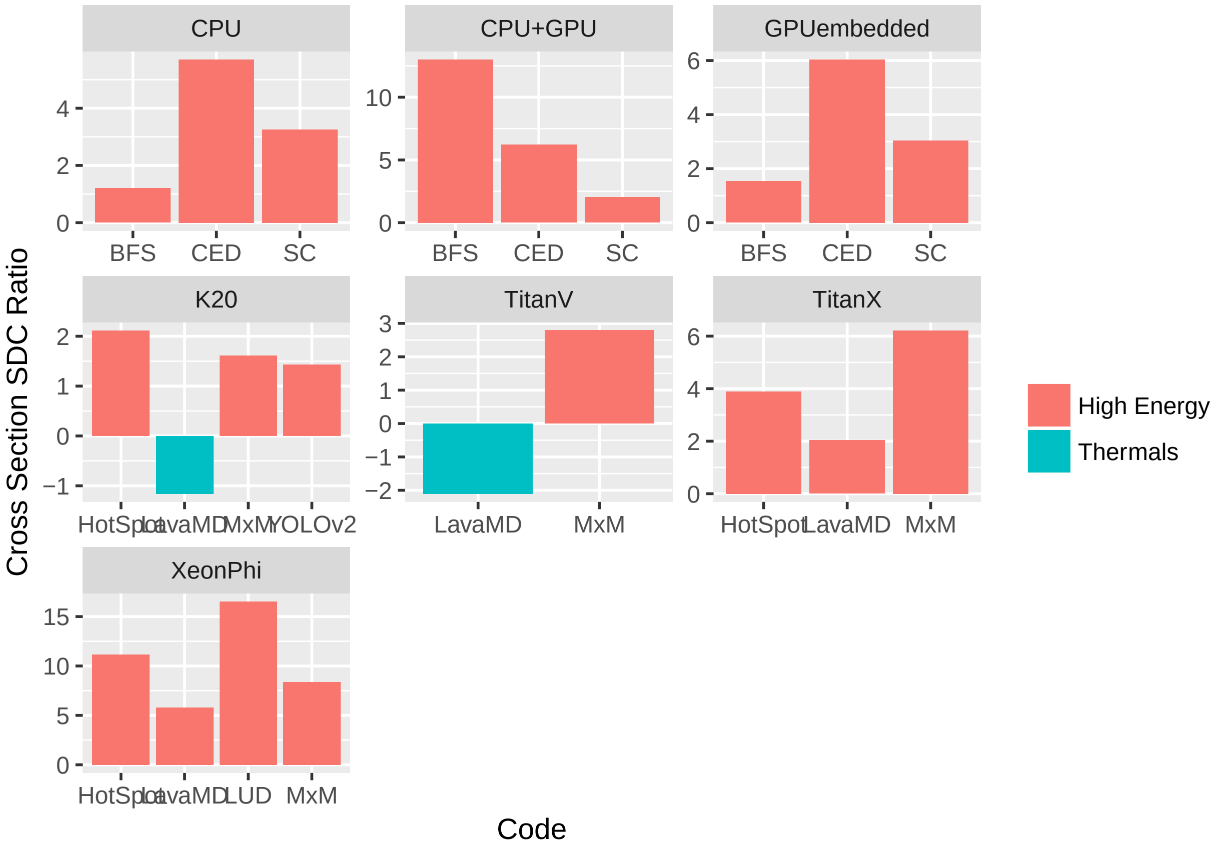
```
cs_due_ratio("CPU+GPU")
```

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



Cross Sections Ratio ALL devices

```
plot = ggplot(datNormalized, aes(x=Code, y=ratioSDC, fill=moreSensitiveSDC)) +
  geom_bar(stat="identity") +
  facet_wrap(~ Device, scales = "free") +
  labs(x = "Code", y = "Cross Section SDC Ratio", fill="") #+
  #theme(legend.position="bottom",axis.text.x=element_text(size=14) , axis.text.y=element_text(size=14)
plot
```



FIT

Functions to plot FIT data

```
fit_sdc_plot_errbars <- function(device){
  datPlot = datNormalized[datNormalized$Device == device, ]

  dfm1 = melt(datPlot[, c('Code', 'FIT_SDC_high_err', 'FIT_SDC_high')], id.vars = c(1, 2))
  colnames(dfm1) = c("Code", "Error", "Neutron_Type", "FIT")
  dfm1$Neutron_Type = "High Energy"
  dfm2 = melt(datPlot[, c('Code', 'FIT_SDC_thermals_err', 'FIT_SDC_thermals')], id.vars = c(1, 2))
  colnames(dfm2) = c("Code", "Error", "Neutron_Type", "FIT")
  dfm2$Neutron_Type = "Thermals"
  dfm3 = melt(datPlot[, c('Code', 'FIT_SDC_thermals_storm_err', 'FIT_SDC_thermals_storm')], id.vars = c
  colnames(dfm3) = c("Code", "Error", "Neutron_Type", "FIT")
  dfm3$Neutron_Type = "Thermals (Storm)"
  dfm = rbind(dfm1, dfm2, dfm3)

  plot = ggplot(dfm, aes(x=Code, weight=FIT, ymin=FIT-Error, ymax=FIT+Error, fill=Neutron_Type)) +
  geom_bar      (position=position_dodge(), aes(y=FIT), stat="identity") +
  geom_errorbar (position=position_dodge(width=0.9), width=0.2, colour="black") +
  labs(x = "", y = "FIT SDC [a.u.]", fill="") +
  theme(legend.position="bottom",axis.text.x=element_text(size=14) , axis.text.y=element_text(size=14)
  filename = paste("./plots/FIT_SDC_", device, ".pdf", sep="")
  ggsave(filename, plot)
  plot
```

```
}

fit_due_plot_errbars <- function(device){
  datPlot = datNormalized[datNormalized$Device == device, ]

  dfm1 = melt(datPlot[, c('Code', 'FIT_DUE_high_err', 'FIT_DUE_high')], id.vars = c(1, 2))
  colnames(dfm1) = c("Code", "Error", "Neutron_Type", "FIT")
  dfm1$Neutron_Type = "High Energy"
  dfm2 = melt(datPlot[, c('Code', 'FIT_DUE_thermals_err', 'FIT_DUE_thermals')], id.vars = c(1, 2))
  colnames(dfm2) = c("Code", "Error", "Neutron_Type", "FIT")
  dfm2$Neutron_Type = "Thermals"
  dfm3 = melt(datPlot[, c('Code', 'FIT_DUE_thermals_storm_err', 'FIT_DUE_thermals_storm')], id.vars = c
  colnames(dfm3) = c("Code", "Error", "Neutron_Type", "FIT")
  dfm3$Neutron_Type = "Thermals (Storm)"
  dfm = rbind(dfm1, dfm2, dfm3)

  plot = ggplot(dfm, aes(x=Code, weight=FIT, ymin=FIT-Error, ymax=FIT+Error, fill=Neutron_Type)) +
  geom_bar      (position=position_dodge(), aes(y=FIT), stat="identity") +
  geom_errorbar (position=position_dodge(width=0.9), width=0.2, colour="black") +
  labs(x = "", y = "FIT DUE [a.u.]", fill="") +
  theme(legend.position="bottom",axis.text.x=element_text(size=14) , axis.text.y=element_text(size=14)
  filename = paste("./plots/FIT_DUE_", device, ".pdf", sep="")
  ggsave(filename, plot)
  plot
}

sdcplot <- function(device){
datPlot = datNorm[datNorm$Device == device, ]
  plot = ggplot(datPlot, aes(x=datPlot$Code, y=datPlot$FITSDC, fill=Neutron_type)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title = device, x = "", y = "FIT SDC [a.u.]", fill="") +
  theme(plot.title = element_text(hjust = 0.5), legend.position="bottom",axis.text.x=element_text(size=
  filename = paste("~/plots/FIT-SDC-", device, ".pdf", sep="")
  ggsave(filename, plot)
  plot
}

dueplot <- function(device){
  datPlot = datNorm[datNorm$Device == device, ]
  plot = ggplot(datPlot, aes(x=datPlot$Code, y=datPlot$FITCrash, fill=Neutron_type)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title = device, x = "", y = "FIT DUE [a.u.]", fill="") +
  theme(plot.title = element_text(hjust = 0.5), legend.position="bottom",axis.text.x=element_text(size=
  filename = paste("~/plots/FIT-DUE-", device, ".pdf", sep="")
  ggsave(filename, plot)
  plot
}
```

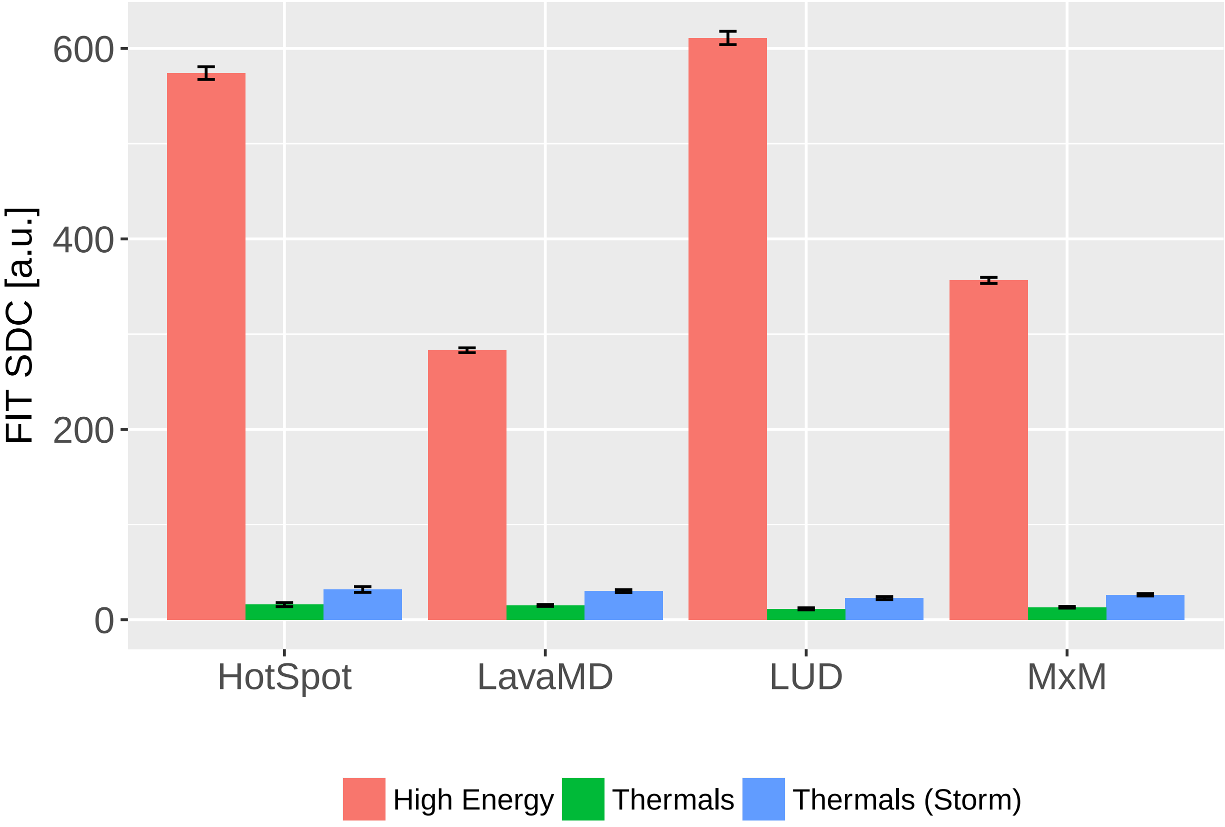
Plots

Xeon Phi

SDC

```
fit_sdc_plot_errbars("XeonPhi")
```

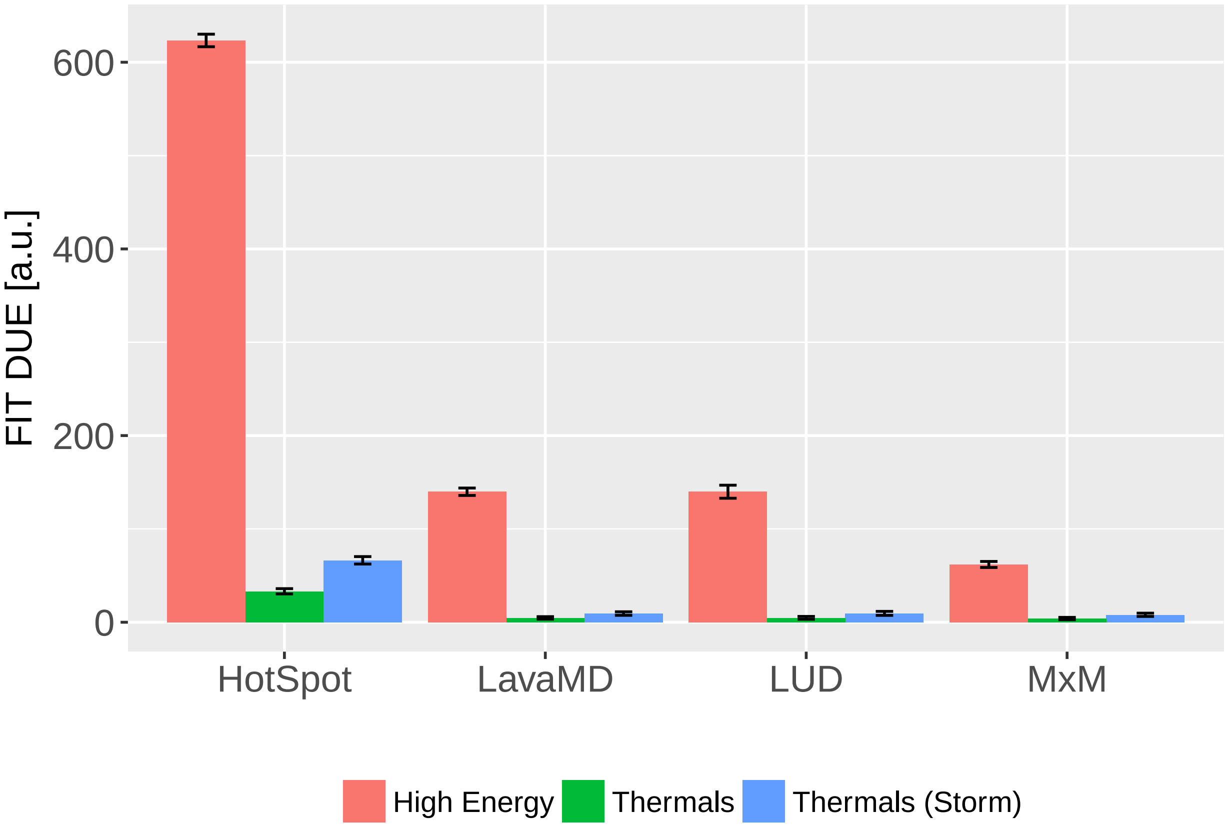
```
## Saving 6.5 x 4.5 in image
```



DUE

```
fit_due_plot_errbars("XeonPhi")
```

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

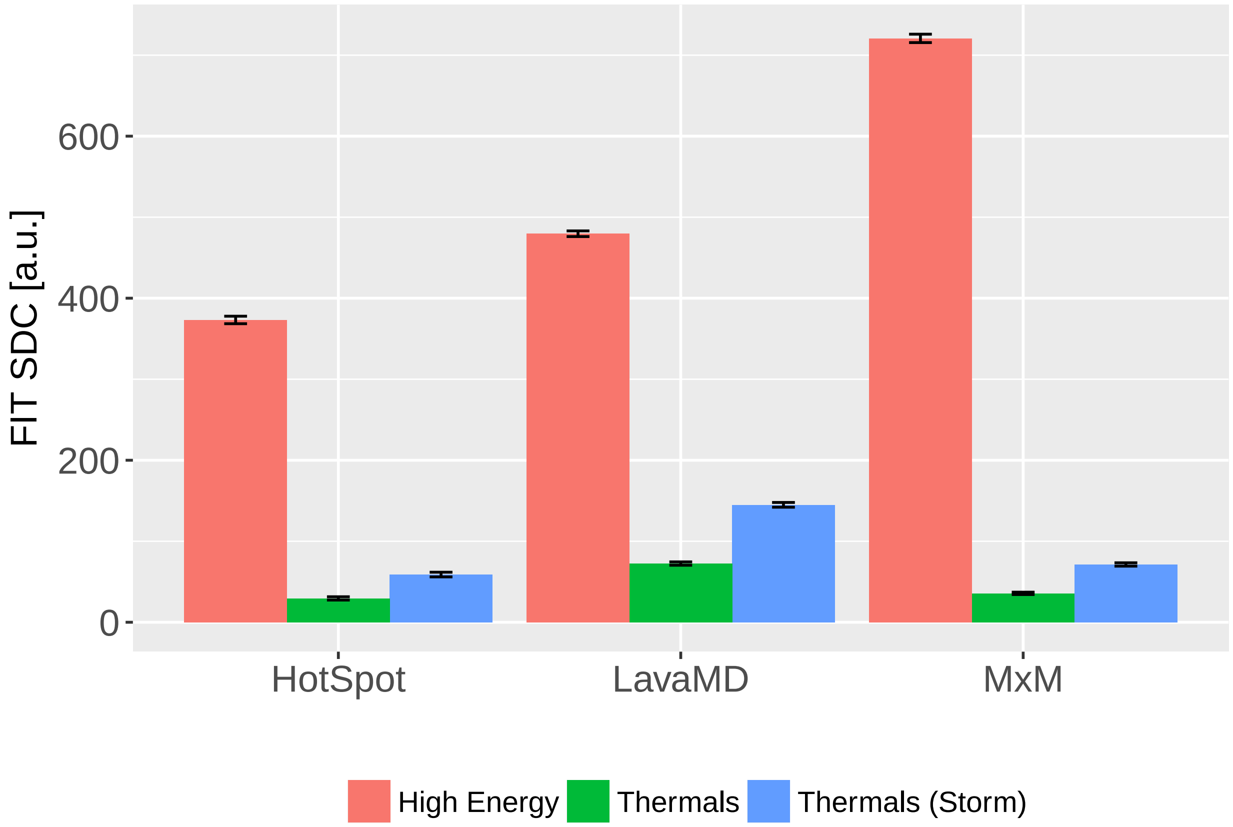


TitanX

SDC

```
fit_sdc_plot_errbars("TitanX")
```

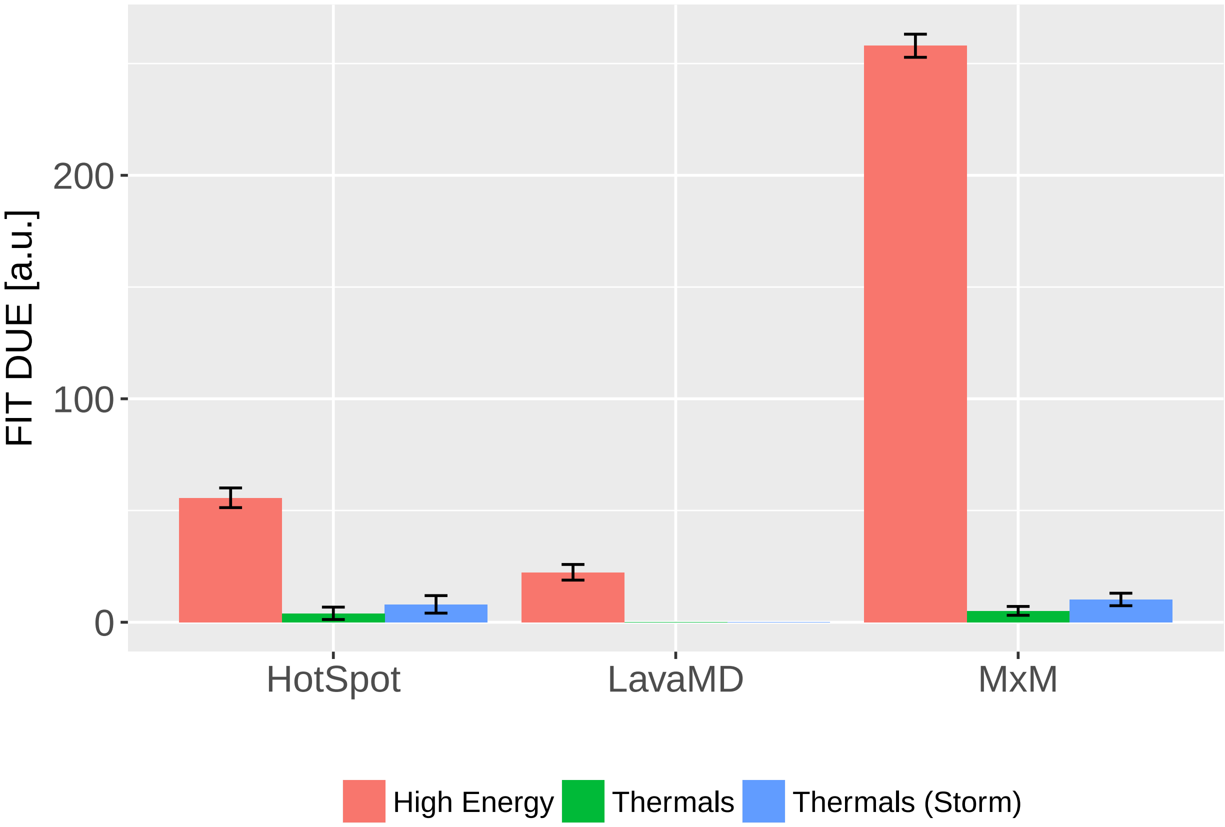
```
## Saving 6.5 x 4.5 in image
```



DUE

```
fit_due_plot_errbars("TitanX")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 2 rows containing missing values (geom_errorbar).

## Warning: Removed 2 rows containing missing values (geom_errorbar).
```

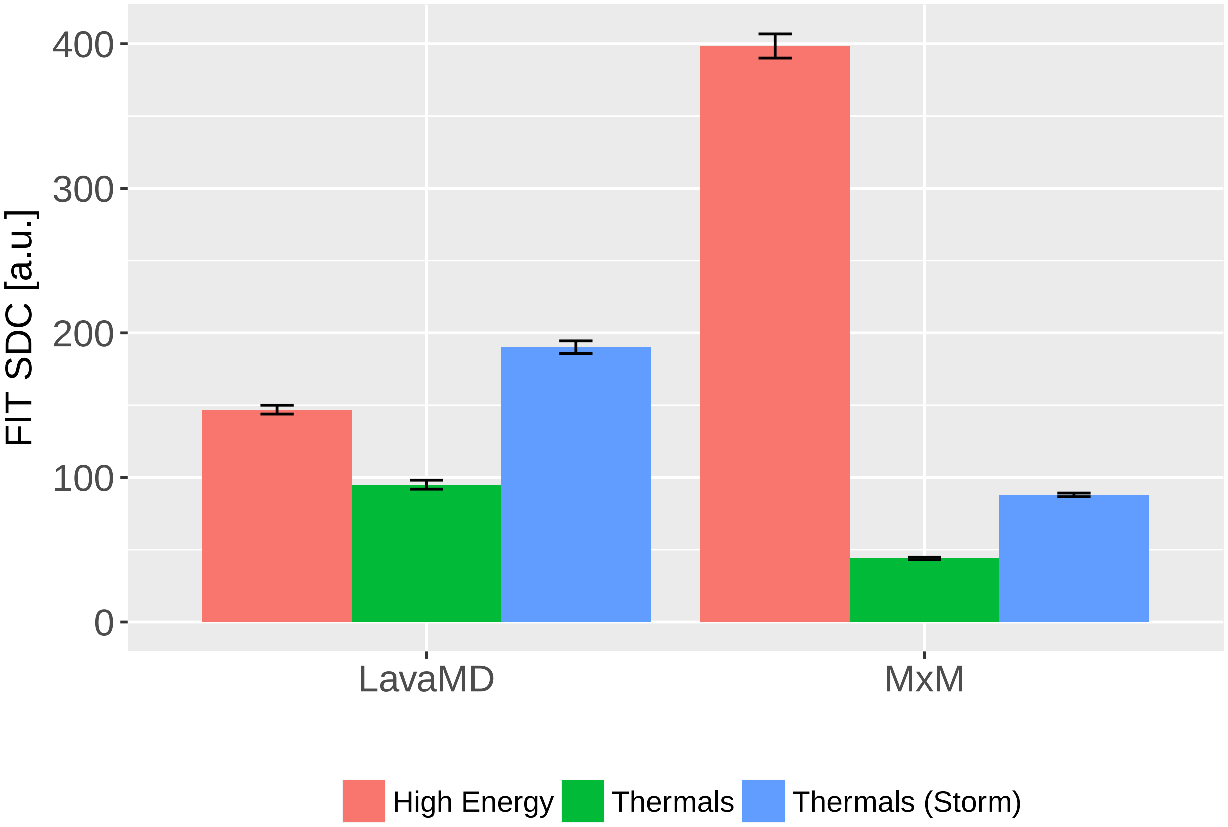


TitanV

SDC

```r
fit_sdc_plot_errbars("TitanV")
```

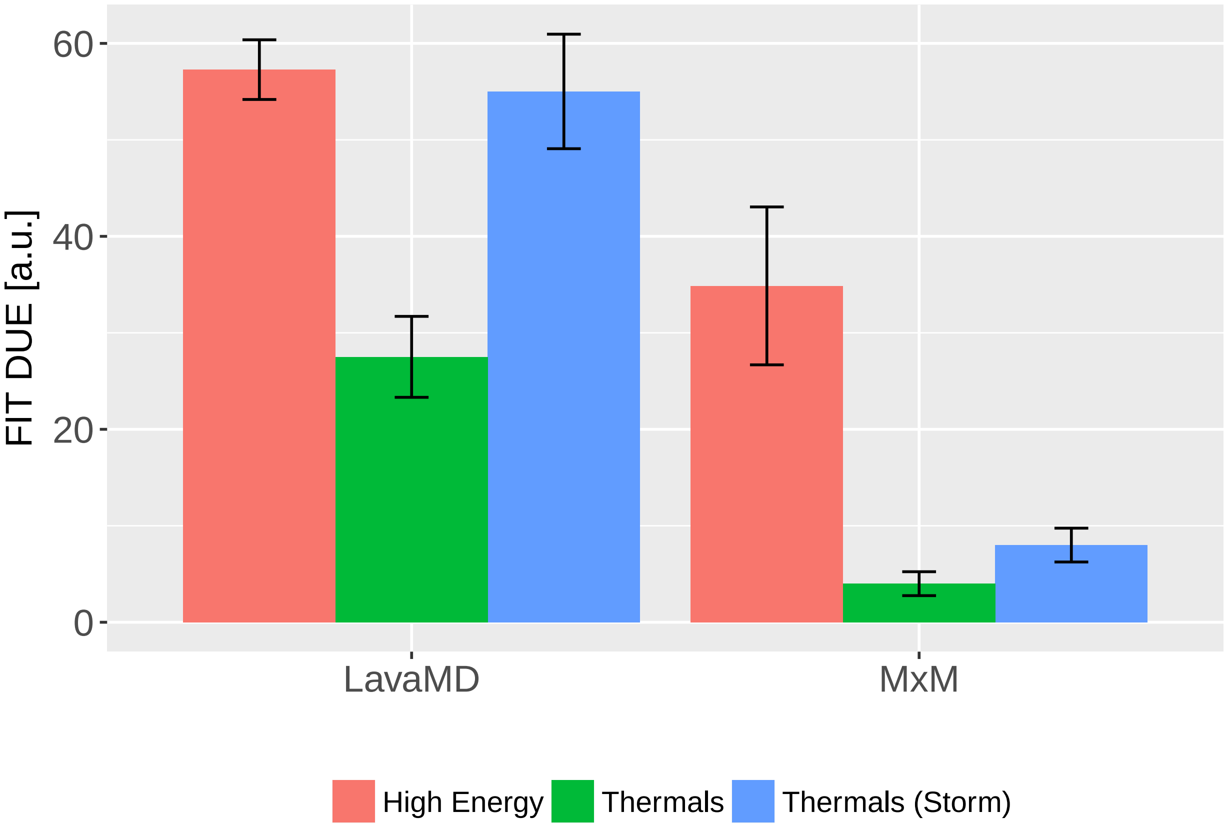
```
## Saving 6.5 x 4.5 in image
```



DUE

```
fit_due_plot_errbars("TitanV")
```

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

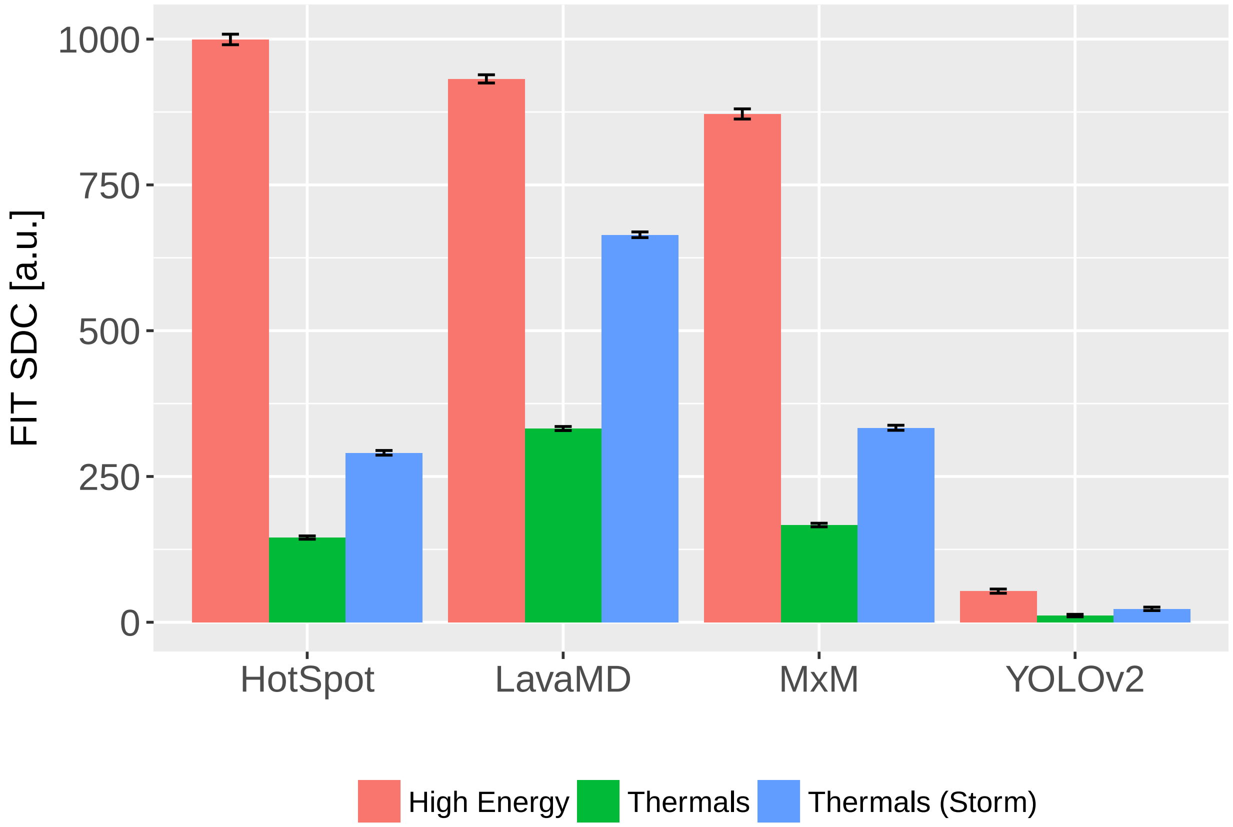


K20

SDC

```
fit_sdc_plot_errbars("K20")
```

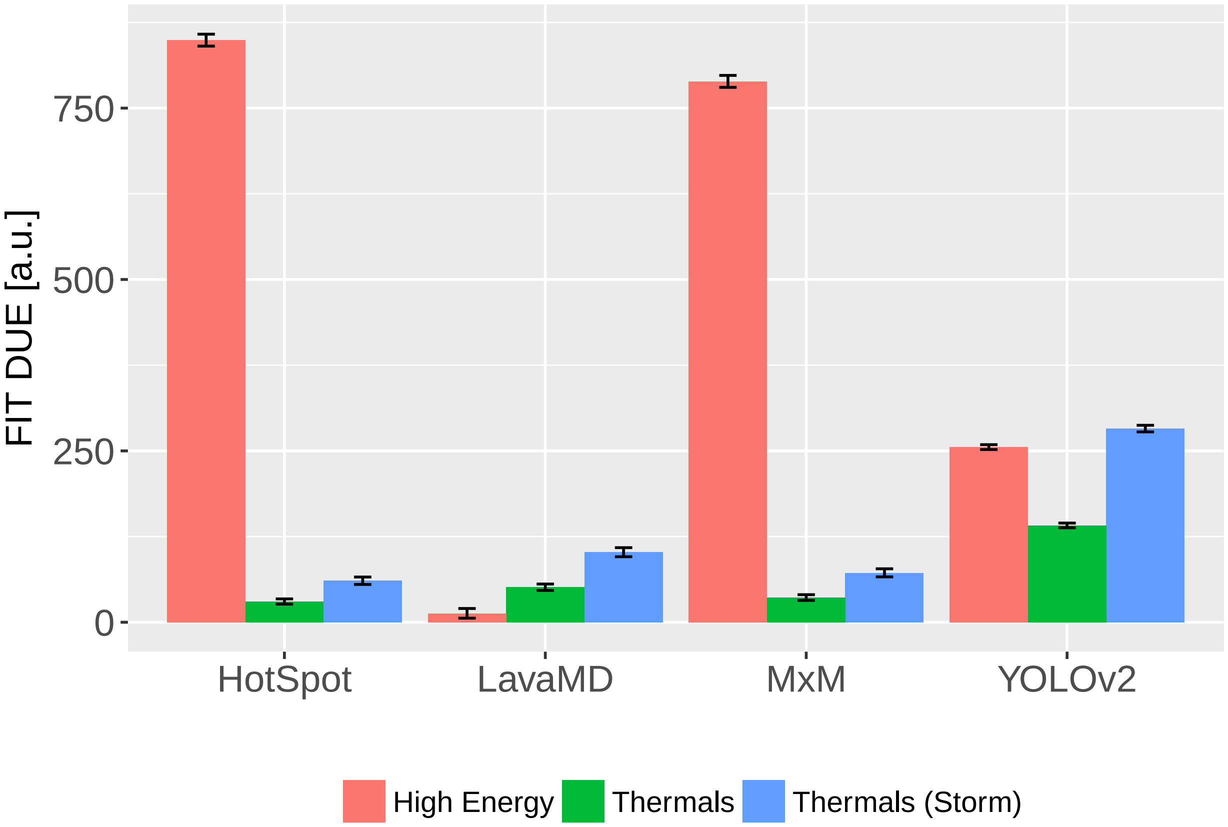
```
## Saving 6.5 x 4.5 in image
```



DUE

```
fit_due_plot_errbars("K20")
```

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

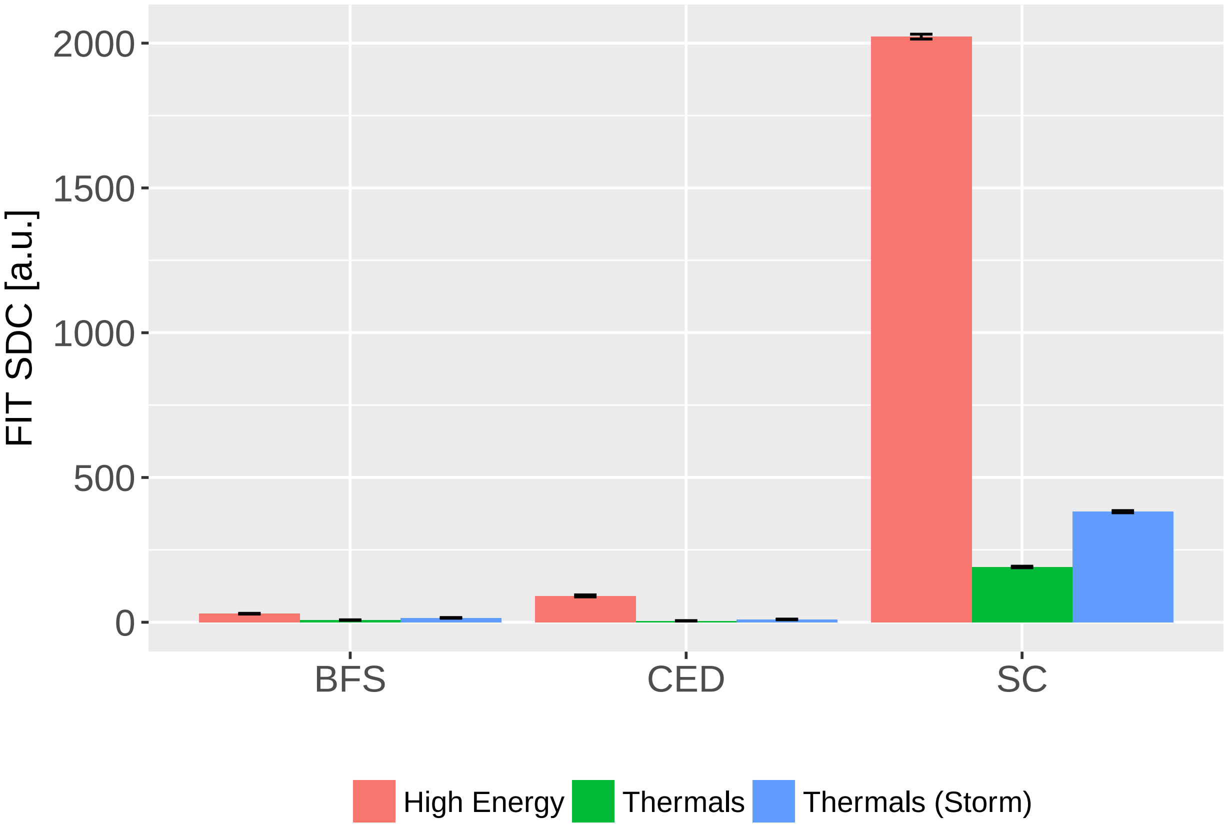


CPU

SDC

```
fit_sdc_plot_errbars("CPU")
```

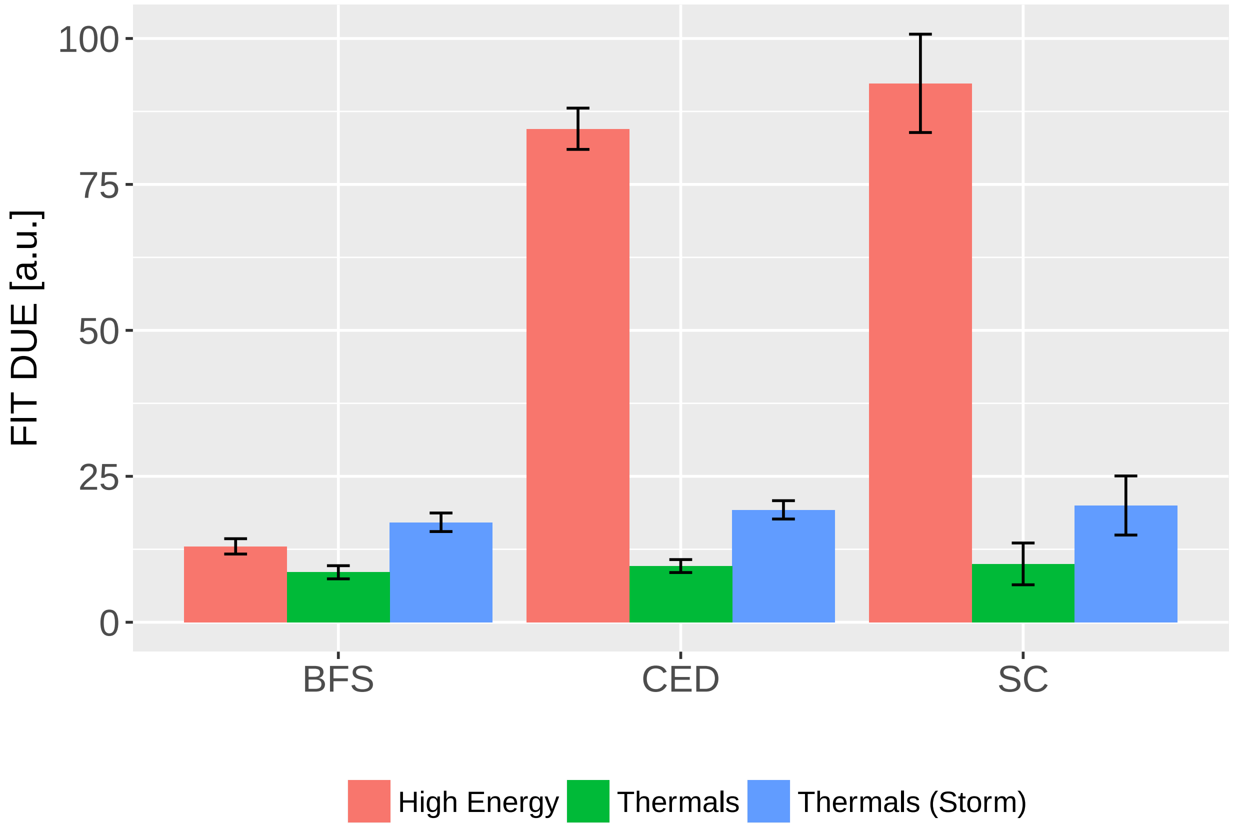
```
## Saving 6.5 x 4.5 in image
```



DUE

```
fit_due_plot_errbars("CPU")
```

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

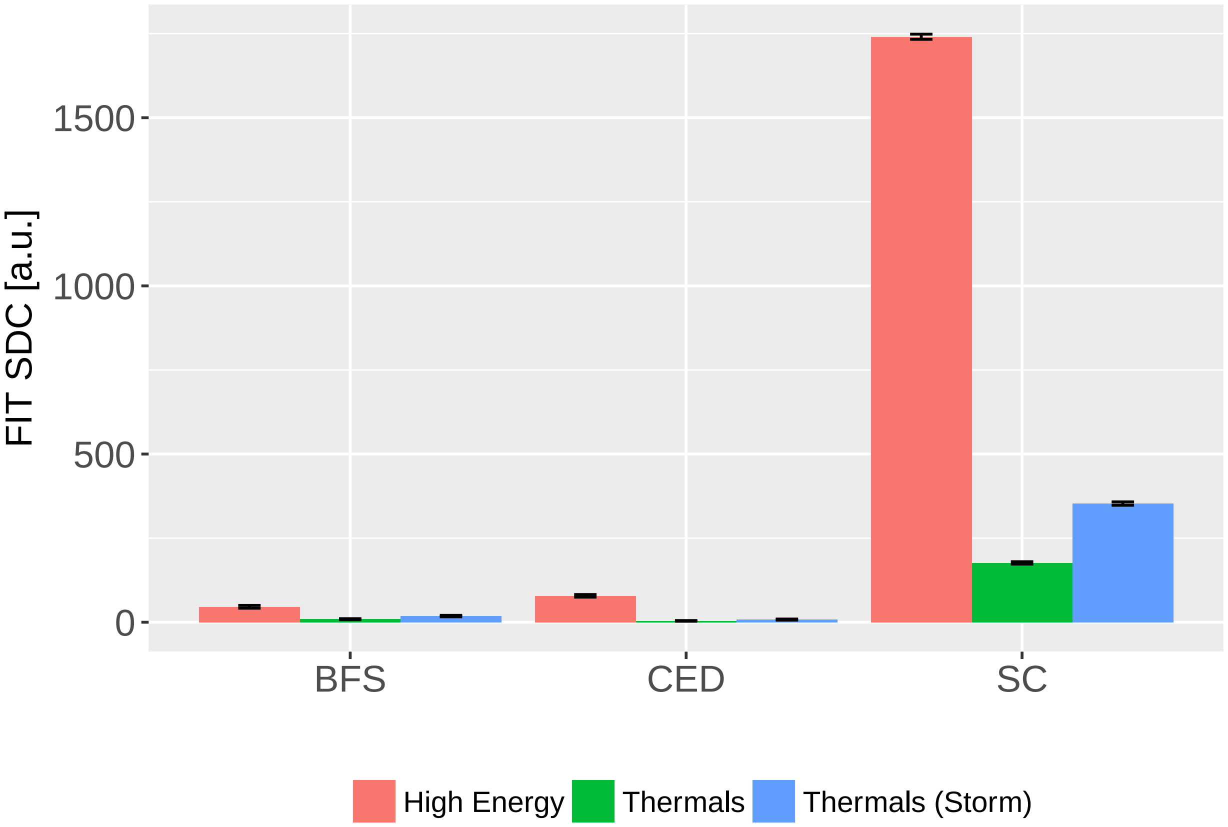


GPUembedded

SDC

```r
fit_sdc_plot_errbars("GPUembedded")
```

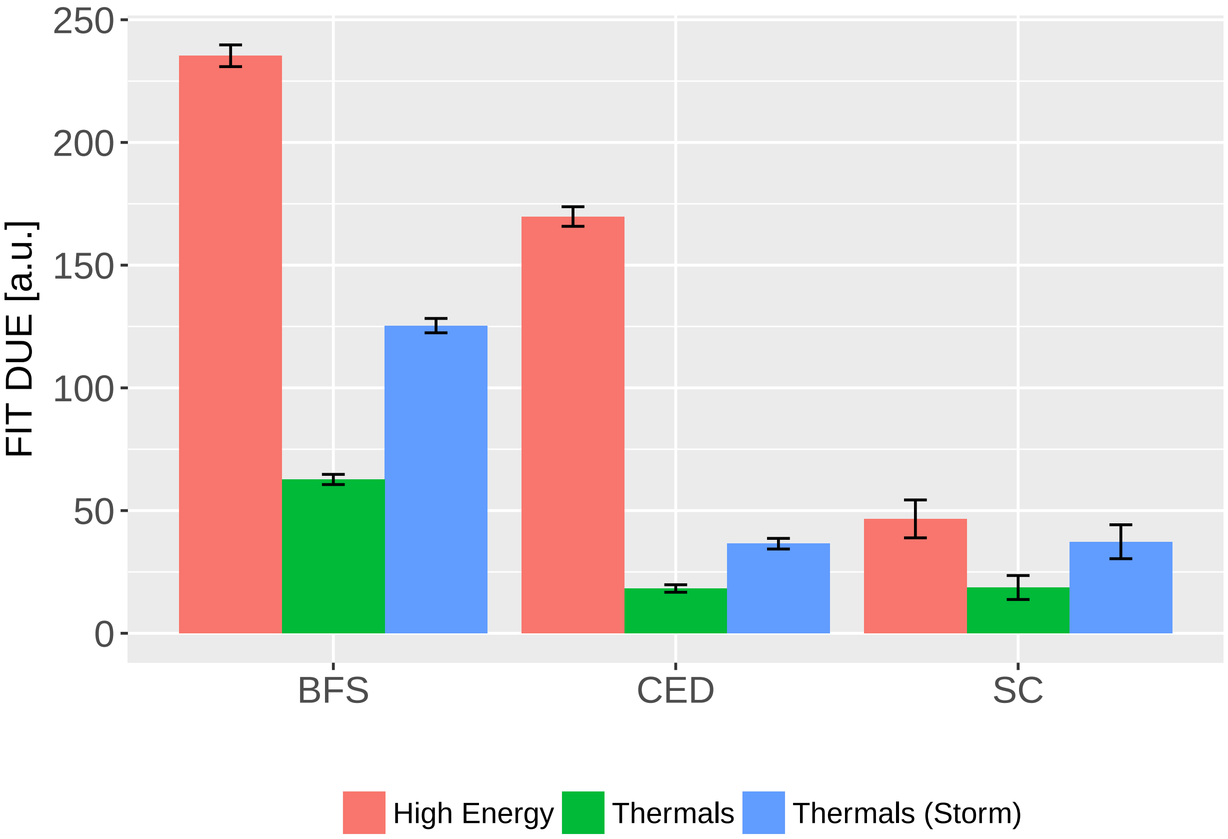
```
## Saving 6.5 x 4.5 in image
```



DUE

```r
fit_due_plot_errbars("GPUembedded")
```

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

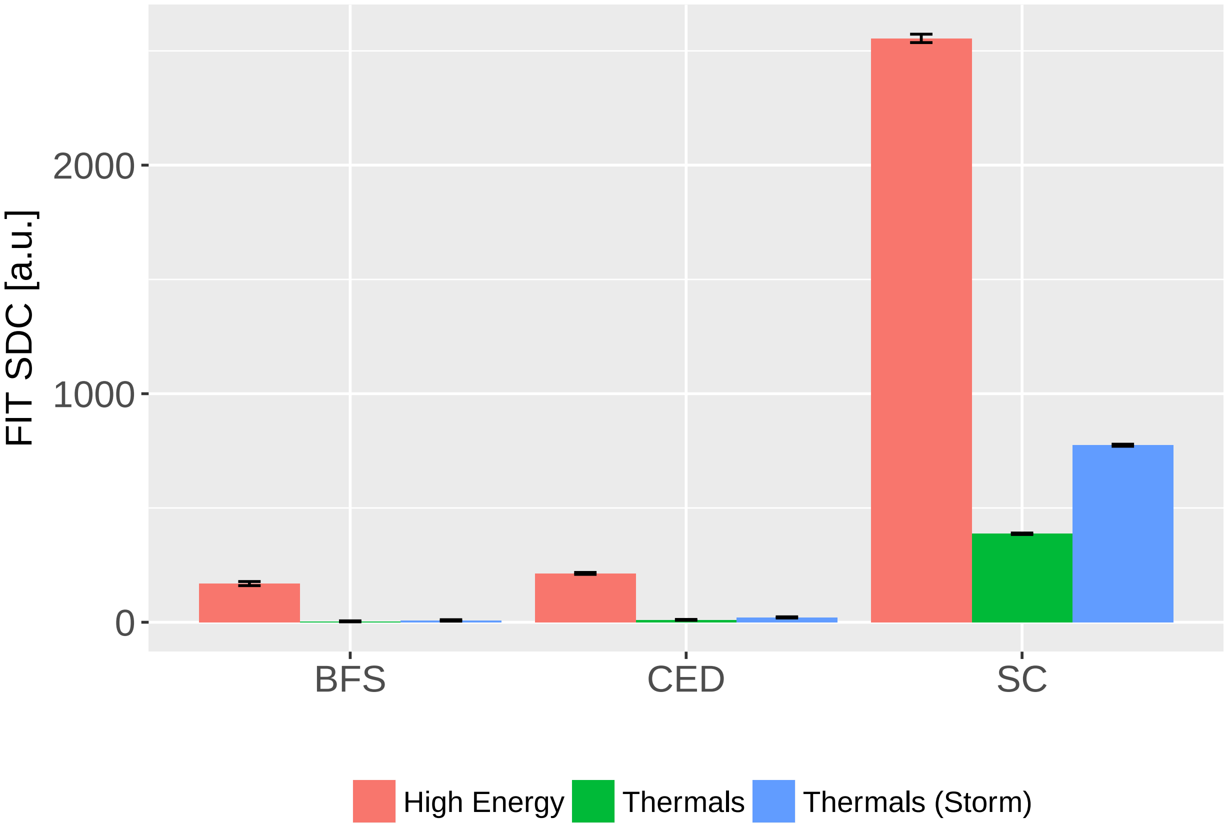


CPU+GPU

SDC

```
fit_sdc_plot_errbars("CPU+GPU")
```

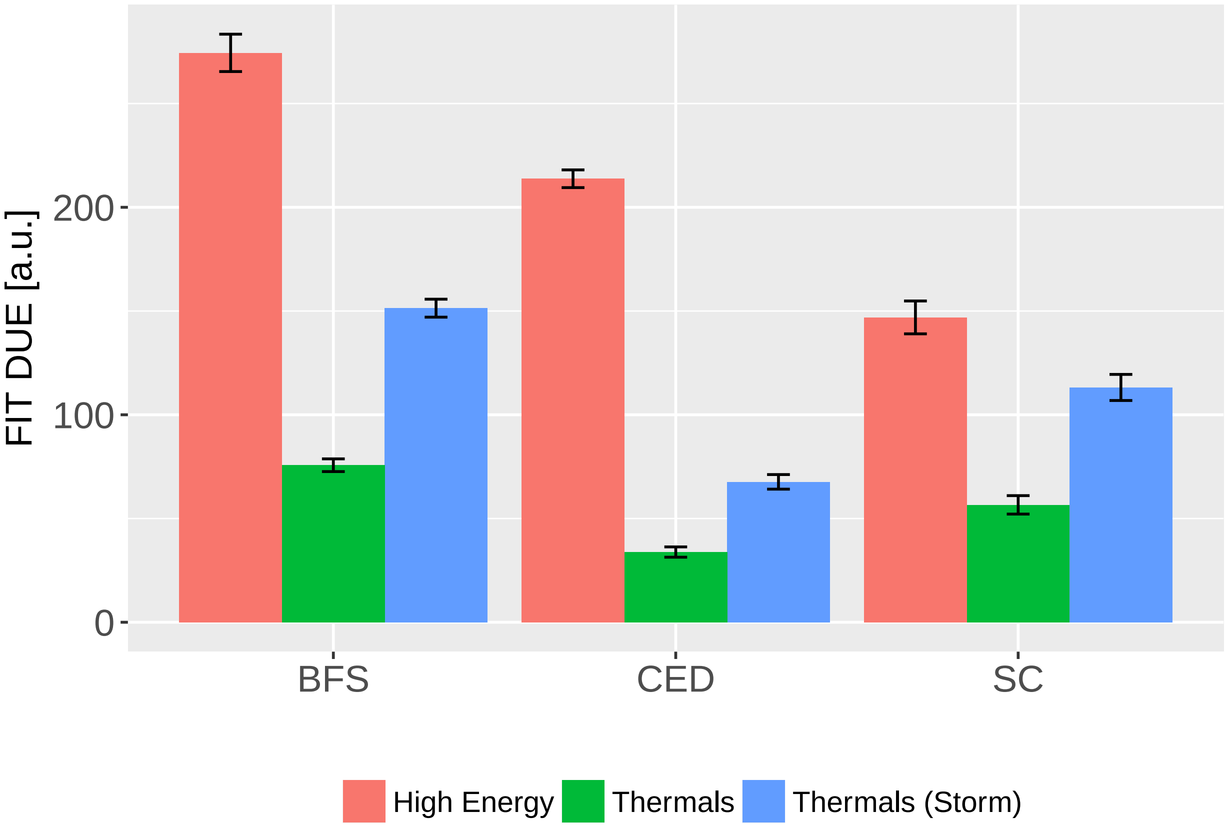
```
## Saving 6.5 x 4.5 in image
```



DUE

```
fit_due_plot_errbars("CPU+GPU")
```

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


FIT DUE [a.u.]
200
100
0
BFS
CED
SC
High Energy
Thermals
Thermals (Storm)