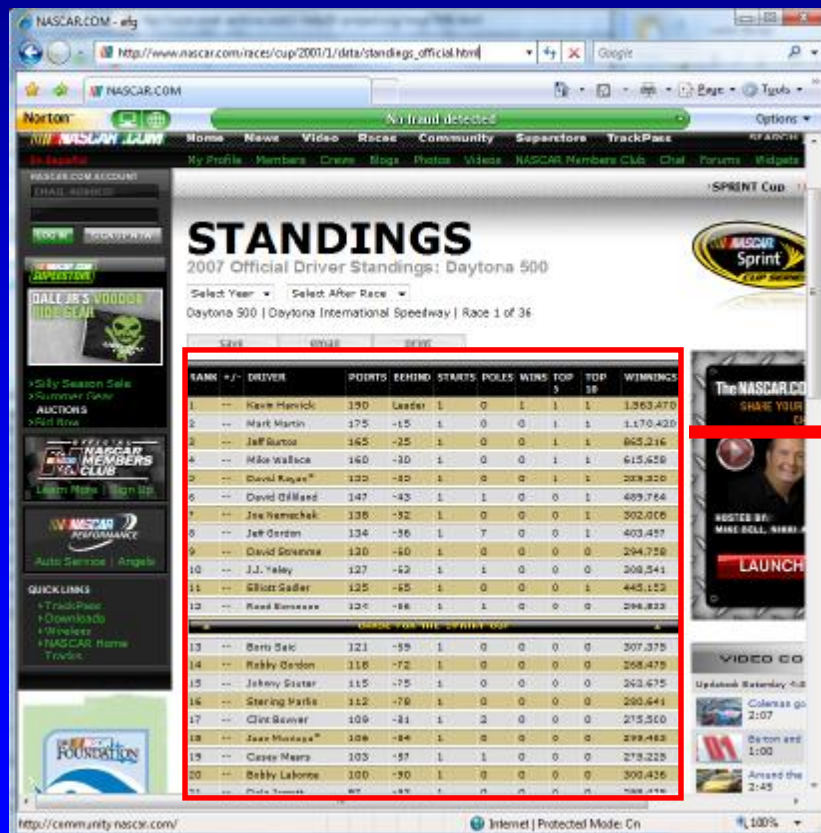


What the original talk was to be about

XML and Rcurl Packages

Extracting a data.frame from HTML code

<http://www.mail-archive.com/r-help@r-project.org/msg17496.html>



STANDINGS
2007 Official Driver Standings: Daytona 500

Select Year: Select After Race:

Daytona 500 | Daytona International Speedway | Race 1 of 36

RANK	DRIVER	POINTS	BEHIND	STARTS	POLES	WINS	TOP 5	TOP 10	WINNINGS
1	Kevin Harvick	190	Leader	1	0	1	1	1	1,563,470
2	Mark Martin	175	-15	1	0	0	1	1	1,170,420
3	Jeff Burton	165	-25	1	0	0	1	1	865,216
4	Mike Wallace	160	-30	1	0	0	1	1	615,658
5	David Ragan*	155	-35	1	0	0	1	1	529,350
6	David Gilliland	147	-43	1	1	0	0	0	489,764
7	Joe Nemechek	138	-52	1	0	0	0	0	302,008
8	Jeff Gordon	134	-56	1	7	0	0	0	403,497
9	David Stremme	130	-60	1	0	0	0	0	294,758
10	J.J. Yeley	127	-63	1	0	0	0	0	308,541
11	Elliott Sadler	125	-65	1	0	0	0	0	445,153
12	Reed Sorenson	124	-66	1	1	0	0	0	296,833
13	Boris Said	121	-69	1	0	0	0	0	307,375
14	Robby Gordon	118	-72	1	0	0	0	0	268,475
15	Johnny Sauter	115	-75	1	0	0	0	0	262,675
16	Sterling Marlin	112	-78	1	0	0	0	0	280,641
17	Clint Bowyer	109	-81	1	2	0	0	0	275,500
18	Juan Montoya*	106	-84	1	0	0	0	0	299,483
19	Casey Mears	103	-87	1	1	0	0	0	275,225
20	Bobby Labonte	100	-90	1	0	0	0	0	300,436
21	Dale Jarrett	97	-93	1	0	0	0	0	295,475
22	Kasey Kahne	96	-94	1	2	0	0	0	425,074
23	Kyle Busch	96	-94	1	0	0	0	0	324,858
24	Carl Edwards	94	-96	1	1	0	0	0	273,383
25	Greg Biffle	88	-102	1	1	0	0	0	317,075
26	Ricky Rudd	85	-105	1	0	0	0	0	293,091
27	Martin Truex, Jr.	81	-109	1	1	0	0	0	290,820
28	Denny Hamlin	79	-111	1	1	0	0	0	316,175
29	Jamie McMurray	70	-120	1	1	0	0	0	265,058
30	Dale Earnhardt, Jr.	67	-123	1	1	0	0	0	355,733
31	Tony Raines	64	-126	1	0	0	0	0	262,583
32	Dave Blaney	61	-129	1	1	0	0	0	269,447
33	Ken Schrader	58	-132	1	0	0	0	0	314,189
34	Jeff Green	55	-135	1	0	0	0	0	259,025
35	Ryan Newman	54	-136	1	5	0	0	0	316,133
36	Kurt Busch	50	-140	1	1	0	0	0	365,316
37	Jimmie Johnson	46	-144	1	4	0	0	0	353,386
38	David Reutimann*	43	-147	1	0	0	0	0	249,583
39	Tony Stewart	39	-151	1	0	0	0	0	549,931
40	Matt Kenseth	37	-153	1	0	0	0	0	309,099
41	Kyle Petty	37	-153	1	0	0	0	0	248,050
42	Scott Riggs	27	-163	1	0	0	0	0	304,158
43	Michael Waltrip	27	-163	1	1	0	0	0	269,708

```
...RANK+/-...DRIVER...POINTS...BEHIND...STARTS...POLES...WINS...TOP 5...TOP 10...WINNINGS$
1...1...Kevin Harvick...190...Leader...1...0...1...1...1...1,563,470$
2...2...Mark Martin...175...-15...1...0...0...1...1...1,170,420$
3...3...Jeff Burton...165...-25...1...0...0...1...1...865,216$
4...4...Mike Wallace...160...-30...1...0...0...1...1...615,658$
5...5...David Ragan*...155...-35...1...0...0...1...1...529,350$
6...6...David Gilliland...147...-43...1...1...0...0...0...489,764$
7...7...Joe Nemechek...138...-52...1...0...0...0...0...302,008$
8...8...Jeff Gordon...134...-56...1...7...0...0...0...403,497$
9...9...David Stremme...130...-60...1...0...0...0...0...294,758$
10...10...J.J. Yeley...127...-63...1...0...0...0...0...308,541$
11...11...Elliott Sadler...125...-65...1...0...0...0...0...445,153$
12...12...Reed Sorenson...124...-66...1...1...0...0...0...296,833$
13...13...Boris Said...121...-69...1...0...0...0...0...307,375$
14...14...Robby Gordon...118...-72...1...0...0...0...0...268,475$
15...15...Johnny Sauter...115...-75...1...0...0...0...0...262,675$
16...16...Sterling Marlin...112...-78...1...0...0...0...0...280,641$
17...17...Clint Bowyer...109...-81...1...2...0...0...0...275,500$
18...18...Juan Montoya*...106...-84...1...0...0...0...0...299,483$
19...19...Casey Mears...103...-87...1...1...0...0...0...275,225$
20...20...Bobby Labonte...100...-90...1...0...0...0...0...300,436$
21...21...Dale Jarrett...97...-93...1...0...0...0...0...295,475$
22...22...Kasey Kahne...96...-94...1...2...0...0...0...425,074$
23...23...Kyle Busch...96...-94...1...0...0...0...0...324,858$
24...24...Carl Edwards...94...-96...1...1...0...0...0...273,383$
25...25...Greg Biffle...88...-102...1...1...0...0...0...317,075$
26...26...Ricky Rudd...85...-105...1...0...0...0...0...293,091$
27...27...Martin Truex, Jr...81...-109...1...1...0...0...0...290,820$
28...28...Denny Hamlin...79...-111...1...1...0...0...0...316,175$
29...29...Jamie McMurray...70...-120...1...1...0...0...0...265,058$
30...30...Dale Earnhardt, Jr...67...-123...1...1...0...0...0...355,733$
31...31...Tony Raines...64...-126...1...0...0...0...0...262,583$
32...32...Dave Blaney...61...-129...1...1...0...0...0...269,447$
33...33...Ken Schrader...58...-132...1...0...0...0...0...314,189$
34...34...Jeff Green...55...-135...1...0...0...0...0...259,025$
35...35...Ryan Newman...54...-136...1...5...0...0...0...316,133$
36...36...Kurt Busch...50...-140...1...1...0...0...0...365,316$
37...37...Jimmie Johnson...46...-144...1...4...0...0...0...353,386$
38...38...David Reutimann*...43...-147...1...0...0...0...0...249,583$
39...39...Tony Stewart...39...-151...1...0...0...0...0...549,931$
40...40...Matt Kenseth...37...-153...1...0...0...0...0...309,099$
41...41...Kyle Petty...37...-153...1...0...0...0...0...248,050$
42...42...Scott Riggs...27...-163...1...0...0...0...0...304,158$
43...43...Michael Waltrip...27...-163...1...1...0...0...0...269,708$
```

What the original talk was to be about

XML and RCurl Packages

Extracting a data.frame from HTML code

```
library(XML)
url <- 'http://www.nascar.com/races/cup/2007/1/data/standings_official.html'
xml <- htmlTreeParse(url, useInternal=TRUE)

q <- "//tbody/tr[position() != 13]/td"
dat <- unlist(xpathApply(xml, q, xmlValue))

df <- as.data.frame(t(matrix(dat, 11)))

# Get the headers
q2 <- "//table/tr/th"
dat2 <- unlist(xpathApply(xml, q2, xmlValue))

colnames(df) <- dat2
```

	RANK	+/-	DRIVER	POINTS	BEHIND	STARTS	POLES	WINS	TOP 5	TOP 10	WINNINGS
1	1	--	Kevin Harvick	190	Leader	1	0	1	1	1	1,563,470
2	2	--	Mark Martin	175	-15	1	0	0	1	1	1,170,420
3	3	--	Jeff Burton	165	-25	1	0	0	1	1	865,216
4	4	--	Mike Wallace	160	-30	1	0	0	1	1	615,658
5	5	--	David Ragan*	155	-35	1	0	0	1	1	529,350
6	6	--	David Gilliland	147	-43	1	1	0	0	1	489,764

*I cannot get the Bioconductor package RCurl to work through our proxy server.*²

What the original talk was to be about

XML and RCurl Packages

Extracting a data.frame from HTML code

```
> unlist(xpathApply(xml, "//*", xmlName))
 [1] "html"    "head"    "title"    "meta"    "meta"    "link"    "link"    "link"
 [9] "link"    "script"  "script"   "body"    "div"     "script"  "style"   "script"
[17] "script"  "script"  "div"      "div"     "div"     "script"  "map"     "area"
[25] "area"    "area"    "script"   "noscript" "div"     "form"    "input"   "input"
- - -
[945] "a"       "div"     "a"        "div"     "a"       "div"     "a"       "div"
[953] "a"       "a"       "a"        "a"       "a"       "a"       "a"       "a"
[961] "p"       "img"     "p"        "img"     "p"       "img"     "p"       "script"
[969] "script"  "script"  "script"   "script"  "img"     "div"     "img"
```

```
> q3 <- "//tbody/tr/td/a"
> dat3 <- unlist(xpathApply(xml, q3, xmlAttrs))

> dat3
                                     href
"/drivers/dps/kharvick00/cup/index.html"
                                     href
"/drivers/dps/mmartin00/cup/index.html"
                                     href
"/drivers/dps/jburton00/cup/index.html"
                                     href
"/drivers/dps/mwallace00/cup/index.html"
. . .
```

Analyzing Olfactory Response Data in ABF Files

Stowers Institute for Medical Research
R/Bioconductor Discussion Group

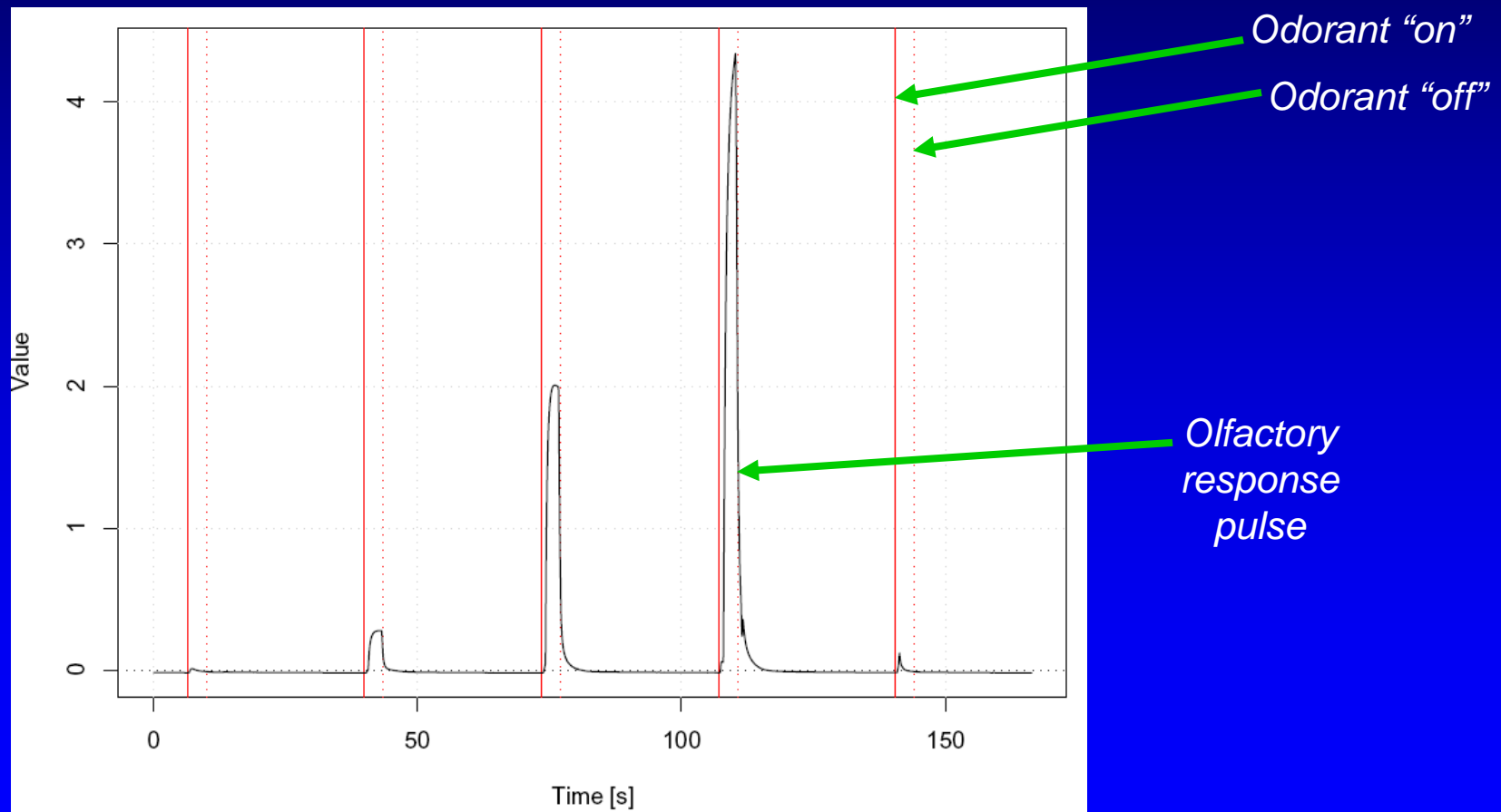
Earl F. Glynn
Scientific Programmer
25 Sept 2008

Analyzing ABF Files

- Converting Axon Binary File to CSV
- Processing directory of ABF files in R
- Finding events in TTL channel
- Finding features and areas in olfactory response "Value" channel
- Reporting results in file and charts

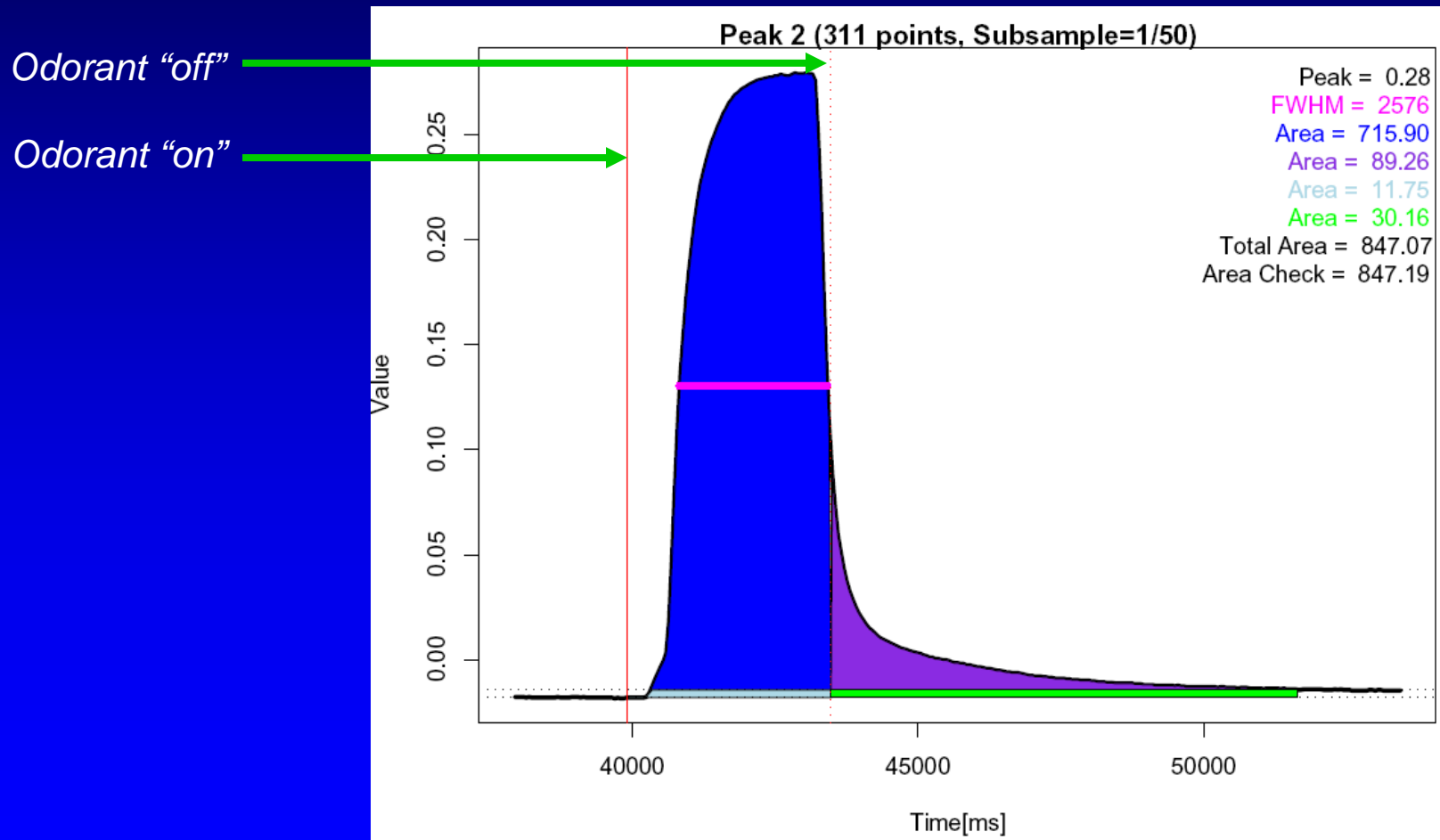
Analyzing ABF Files

U:\efg\Research\RonYu\Limei\R\071225Z-AA-200cm-3s.pdf



Analyzing ABF Files

U:\efg\Research\RonYu\Limei\R\071225Z-AA-200cm-3s.pdf



Olfactory response pulse

Converting Axon Binary File to CSV

U:\efg\Research\RonYu\ABF\Converting-abf-to-csv.doc

- Developed "C" program to extract data from binary ABF file
- Only process "Gap free files" at present
- Only process needed subset of data
- Requires proprietary abffio.dll at run-time
- CSV files are ~4X larger than ABF files
- Many programs can read CSVs;
few programs can read ABFs

Converting Axon Binary File to CSV

U:\efg\Research\RonYu\Limei\071225-Length

Use `system` to run command-line program

e.g., `abf2csv 071225A-AA-25cm-1s.abf 071225A-AA-25cm-1s.csv`

```
read.abf.file <- function(abffile, delete.csv=TRUE)
{
  basedir <- substr(abffile,1,nchar(abffile)-3)
  csvfile <- paste(basedir, "csv", sep="")

  system(paste("abf2csv", abffile, csvfile))

  raw <- read.csv(csvfile, as.is=TRUE)
  colnames(raw) <- c("Value", "TTL")
  raw$Time <- 0:(nrow(raw)-1)

  if (delete.csv)
  {
    file.remove(csvfile)
  }

  return(raw)
}
```

CSV files from ABFs can have too many lines for Excel: e.g., 176,377 in the case above⁹

Analyzing ABF Files

Processing directory of files in R

U:\efg\Research\RonYu\Limei\R\ComputeAreas.R

```
pick.and.process.directory <- function()  
{  
  abf.folder <- choose.dir(default = "U:/efg/Research/RonYu/Limei/",  
                           caption = "Select .abf folder")  
  process.directory(abf.folder)  
}
```



Processing directory of ABF files in R

```
process.directory <- function(abf.folder)
{
  abflist <- dir(path=abf.folder, pattern=".abf$")
  . . .
  for (file.index in 1:length(abflist))
  {
    abffile <- file.path(abf.folder, abflist[file.index])
    . . .
  }
  . . .
}
```

Analyzing ABF Files

Processing directory of files in R

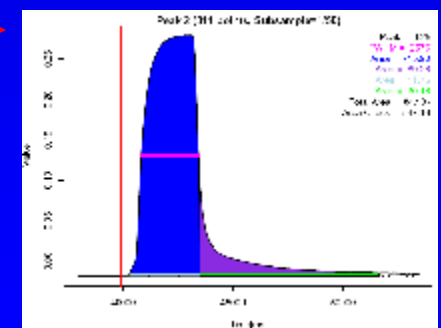
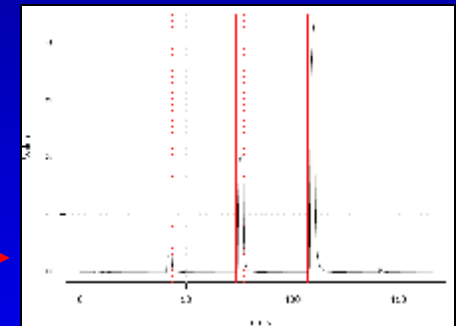
```
process.directory <- function(abf.folder)
{
  abflist <- dir(path=abf.folder, pattern="\\.abf$")
  basefile <- substr(abflist,1,nchar(abflist)-3)
  pdflist <- paste(basefile, "pdf", sep="")
  . . .

  for (file.index in 1:length(abflist))
  {
    abffile <- file.path(abf.folder, abflist[file.index])
    raw <- read.abf.file(abffile)
    TTL <- process.TTL.data(raw)

    subsample <- subsample.raw.data(raw, SUBSAMPLE.FACTOR)

    pdf(file.path(abf.folder, pdflist[file.index]), width=8, height=10)
    par(oma=c(2,0,3,0))
    plot.subsample(subsample, TTL)
    plot.header.and.footer(abffile)

    for (i in 1:length(TTL$start))
    {
      peak.results <- plot.peak(abffile, i, subsample, TTL)
      plot.header.and.footer(abffile)
    }
  }
  dev.off()
}
```



Analyzing ABF Files

Processing directory of files in R

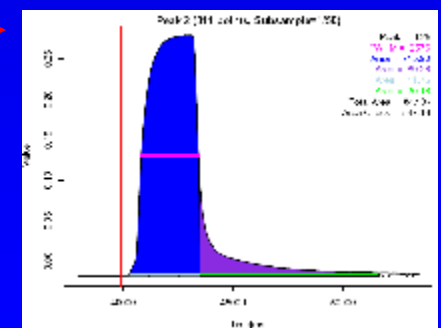
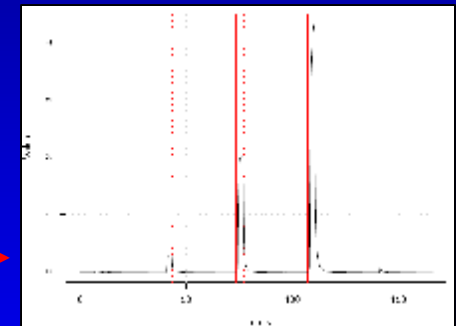
```
process.directory <- function(abf.folder)
{
  abflist <- dir(path=abf.folder, pattern="\\.abf$")
  basefile <- substr(abflist,1,nchar(abflist)-3)
  pdflist <- paste(basefile, "pdf", sep="")
  . . .

  for (file.index in 1:length(abflist))
  {
    abffile <- file.path(abf.folder, abflist[file.index])
    raw <- read.abf.file(abffile)
    TTL <- process.TTL.data(raw)

    subsample <- subsample.raw.data(raw, SUBSAMPLE.FACTOR)

    pdf(file.path(abf.folder, pdflist[file.index]), width=8, height=10)
    par(oma=c(2,0,3,0))
    plot.subsample(subsample, TTL)
    plot.header.and.footer(abffile)

    for (i in 1:length(TTL$start))
    {
      peak.results <- plot.peak(abffile, i, subsample, TTL)
      plot.header.and.footer(abffile)
    }
  }
  dev.off()
}
```



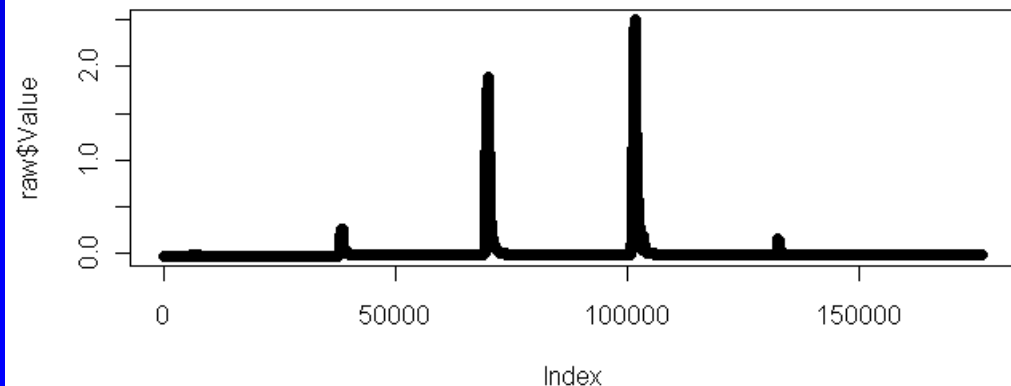
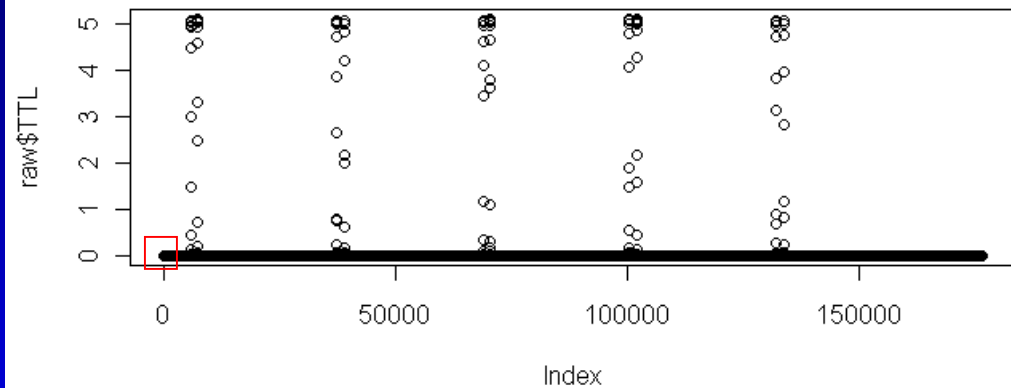
Analyzing ABF Files

Finding events in TTL channel

```
raw <- read.abf.file(abffile)
```

```
par(mfrow=c(2,1))  
plot(raw$TTL)  
plot(raw$Value)
```

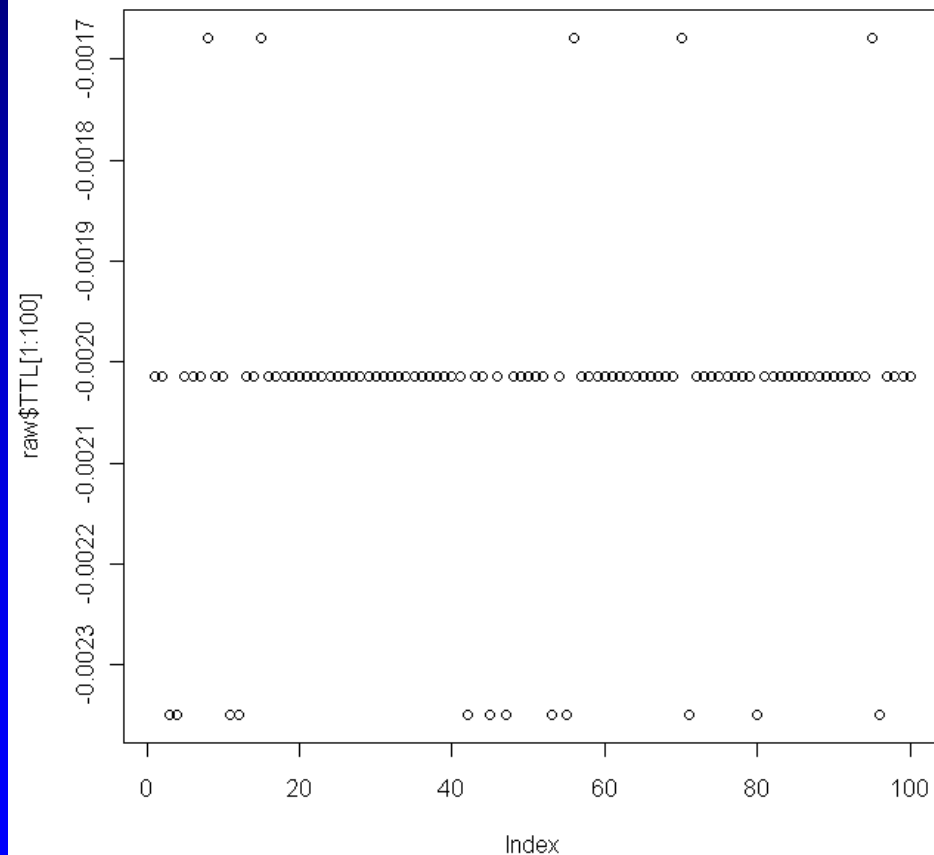
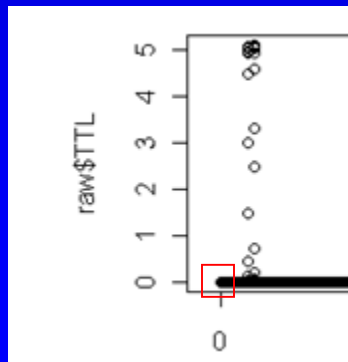
Use TTL events
to process peaks in
olfactory response
"Value" data



Analyzing ABF Files

Finding events in TTL channel

```
plot(raw$TTL[1:100])
```



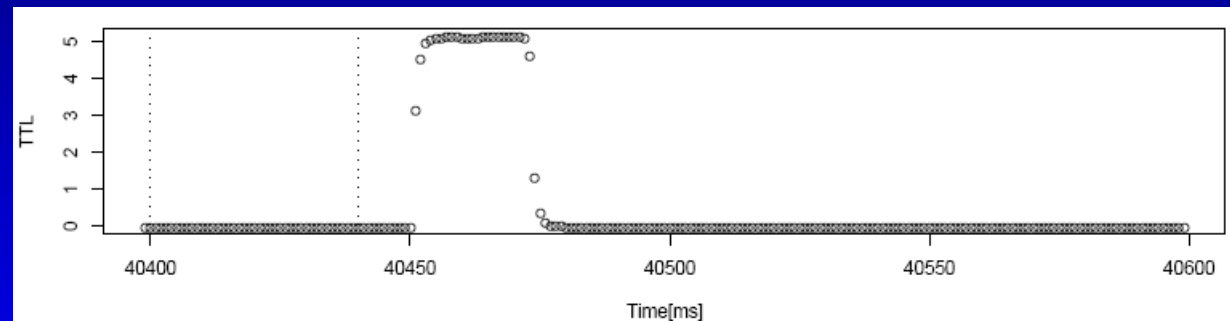
Analyzing ABF Files

Finding events in TTL channel

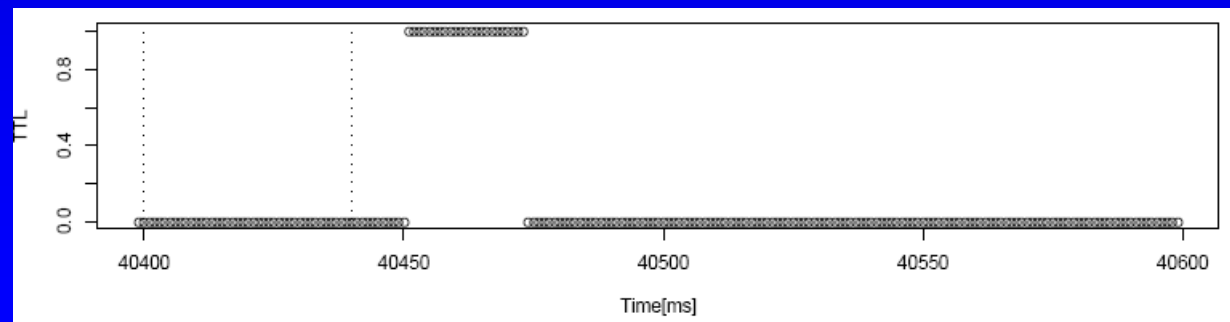
```
TTL.midpoint <- (max(raw$TTL) - min(raw$TTL)) / 2
```

```
fixed.TTL <- ifelse(raw$TTL <= TTL.midpoint, 0, 1)
```

Raw



Fixed



Finding events in TTL channel

Intervals of Interest

```
TTL.delta <- c(diff(fixed.TTL), 0)

TTL.up <- which(TTL.delta == 1)
TTL.up
[1] 5835 7384 37388 38938 68941 70490 100494 102043 132045 133595

TTL.down <- which(TTL.delta == -1)
TTL.down
[1] 5847 7400 37402 38949 68953 70505 100510 102055 132057 133607

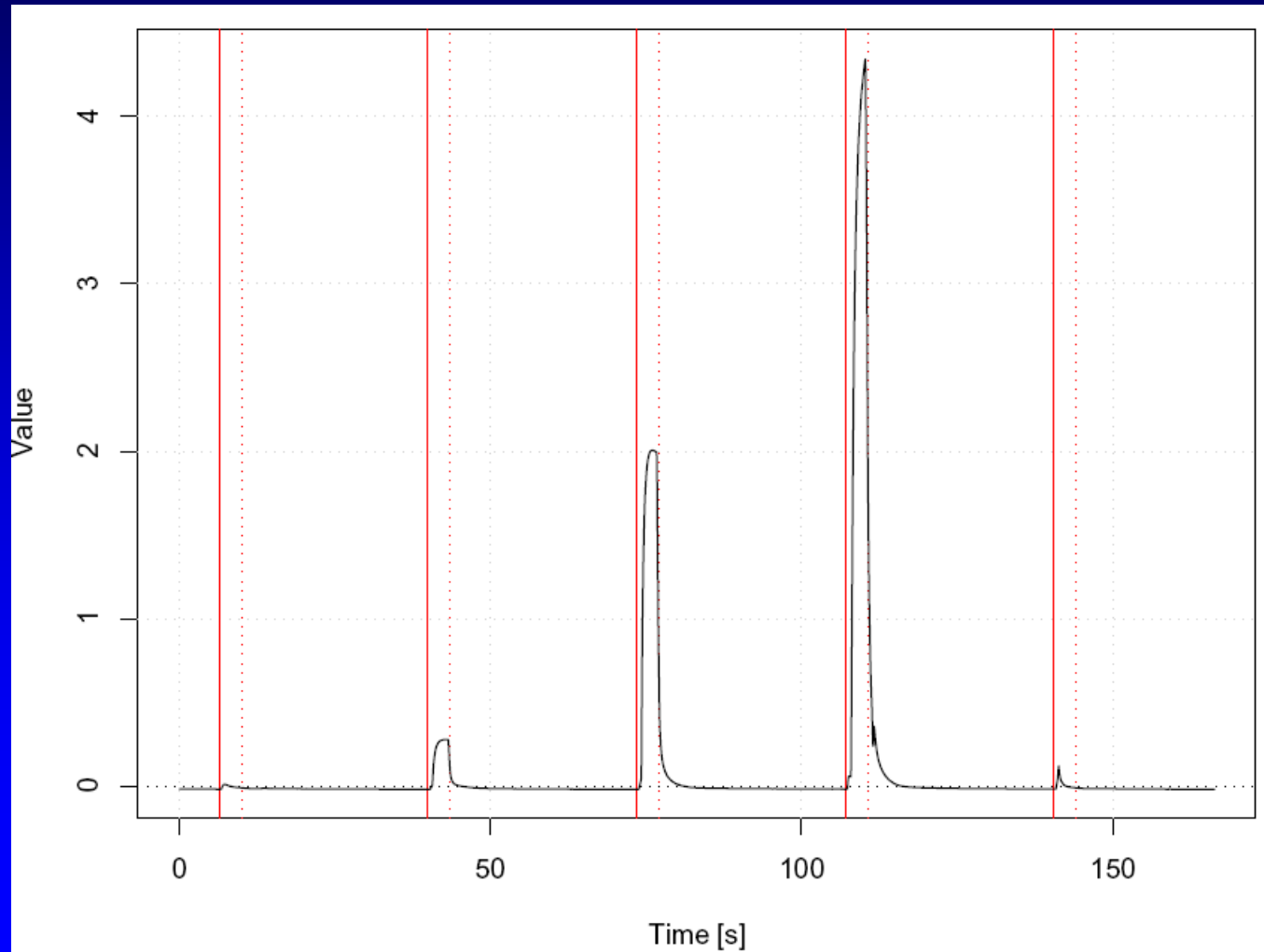
plot(fixed.TTL[5825:5857])
```

```
interval.start <- TTL.up[2*( 1:(length(TTL.up) %/%2) )-1]
interval.start
[1] 5835 37388 68941 100494 132045

interval.stop <- TTL.down[2*( 1:(length(TTL.down) %/%2) ) ]
interval.stop
[1] 7400 38949 70505 102055 133607
```

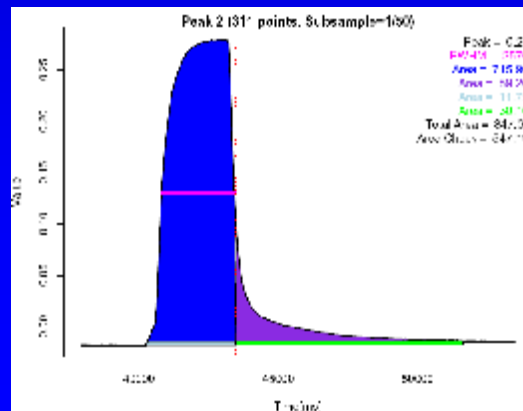


Finding events in TTL channel



Finding features and areas in Value channel

- Need high sampling rate for exact timing of events in TTL channel
- Do not need high sampling rate for most olfactory response features in the Value channel, e.g., area.
- Considerable speedup after subsampling



Subsampling

```
SUBSAMPLE.FACTOR <- 50
```

```
subsample <- subsample.raw.data(raw, SUBSAMPLE.FACTOR)
```

```
subsample.raw.data <- function(raw, subsample.frequency)
{
  # Subsample data -- millisecond resolution too high
  Subsample.Time <- subsample.frequency * 0:((nrow(raw) %/% subsample.frequency) - 1)
  Subsample.Value <- raw$Value[Subsample.Time]

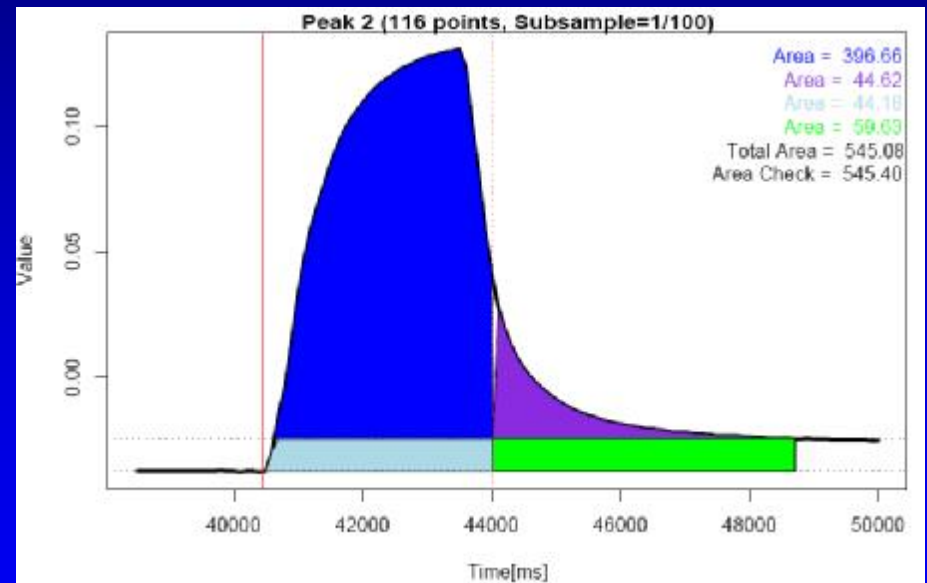
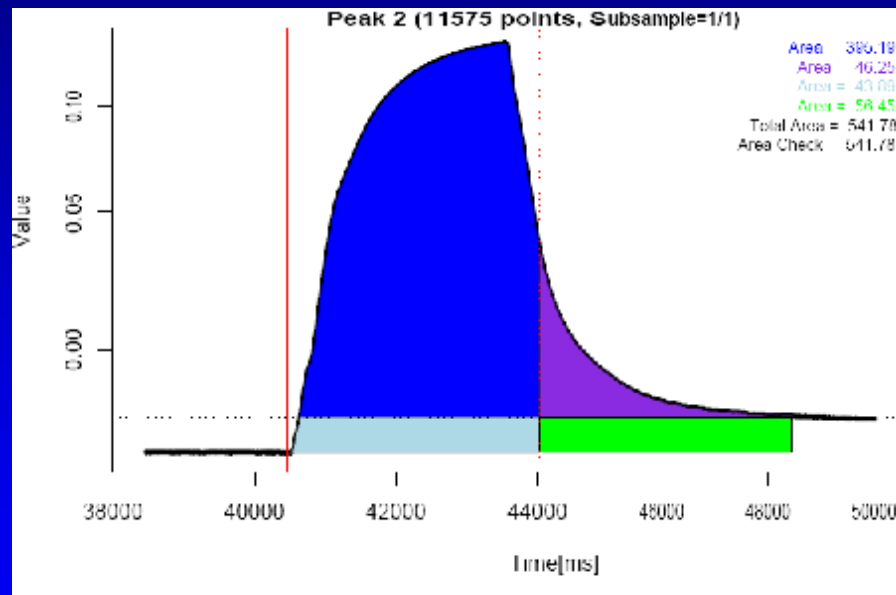
  return(list(Time=Subsample.Time, Value=Subsample.Value, frequency=subsample.frequency))
}
```

Pick every 50th point to speed up processing.
Output PDFs are much smaller with subsampling.

How does Subsampling affect area?

11,575 points

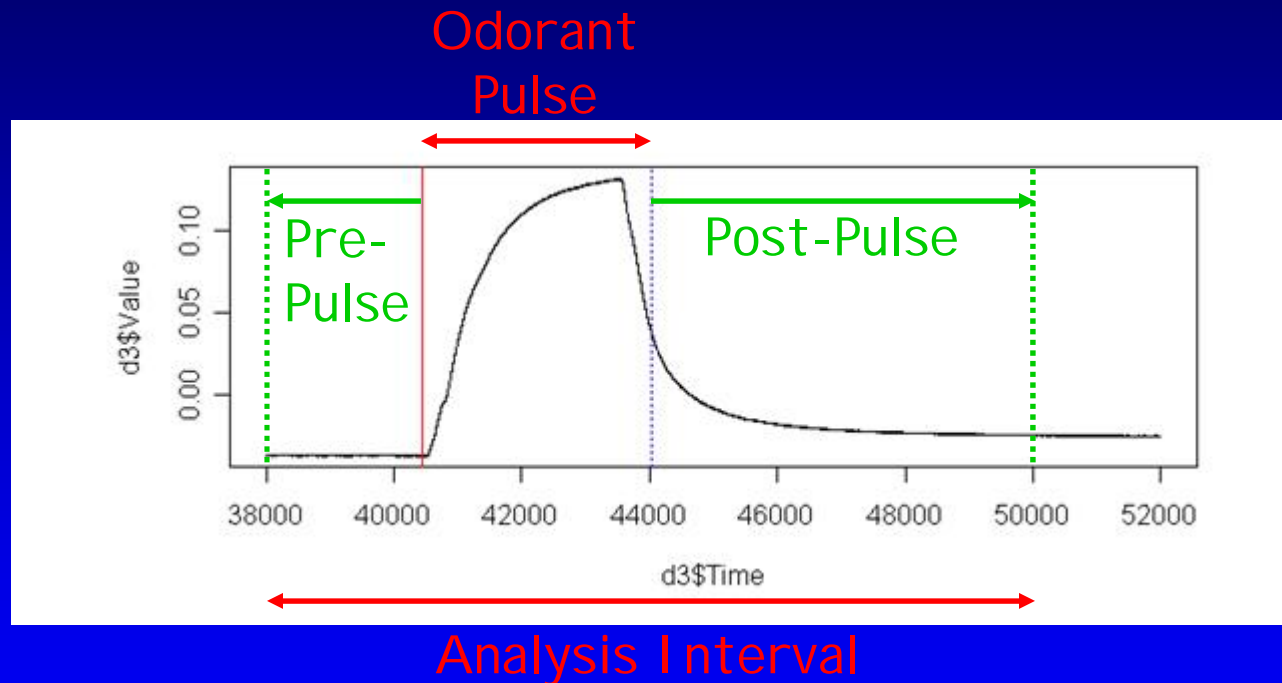
116 points



<1% difference on large areas
up to 6% difference on small areas

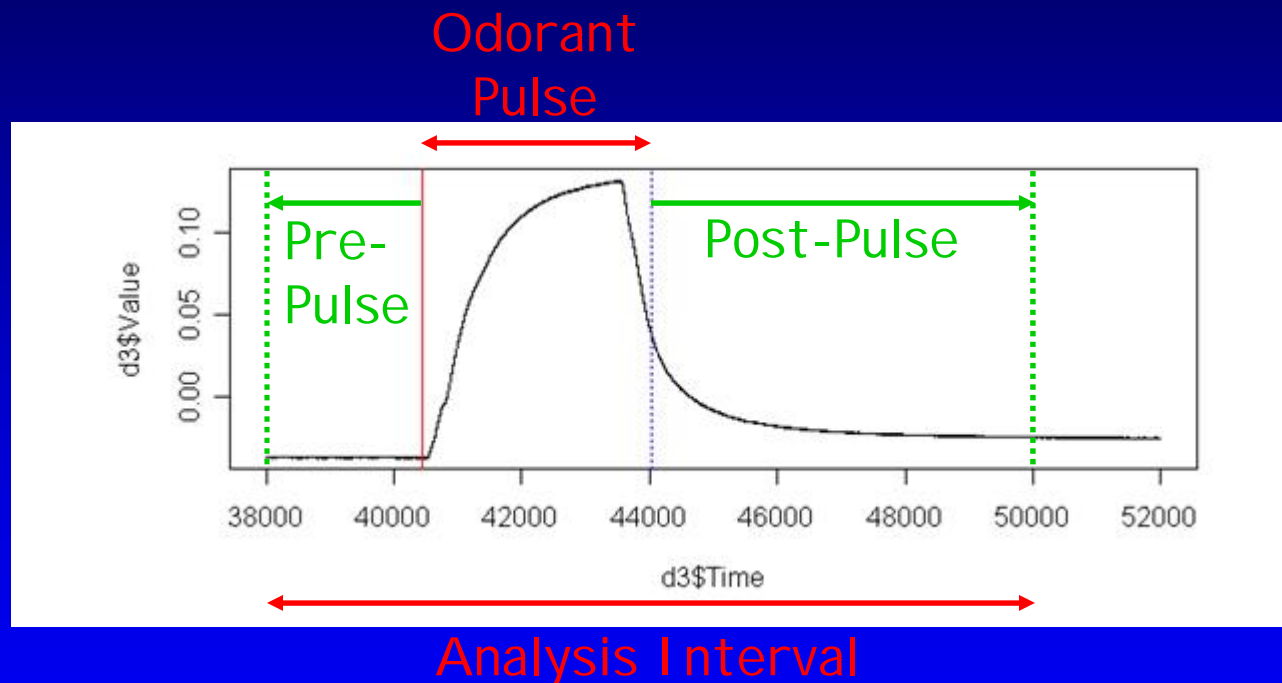
Analyzing ABF Files

Finding features and areas in Value channel



Analyzing ABF Files

Finding features and areas in Value channel



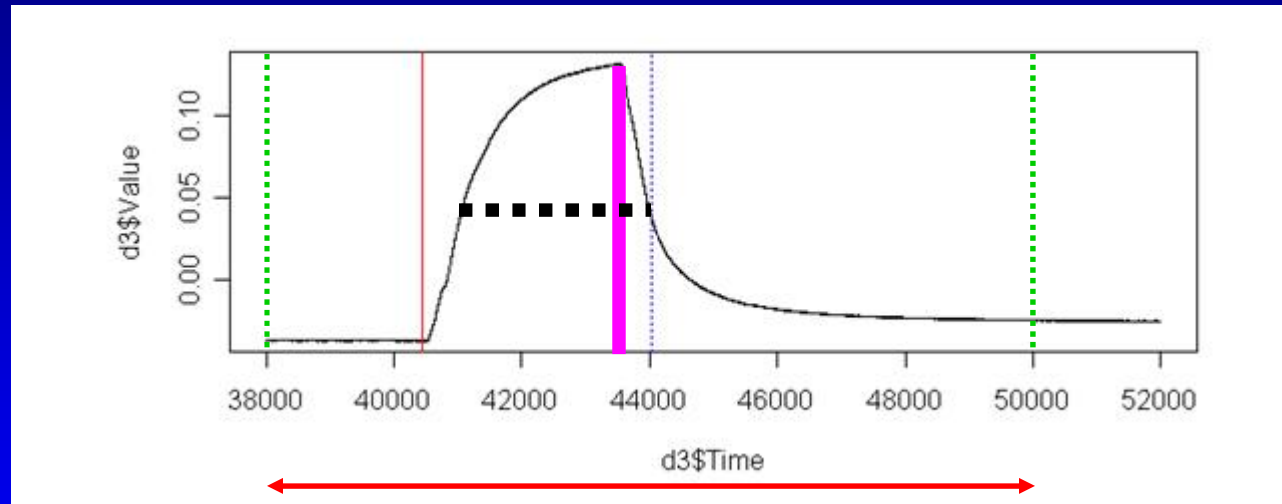
```
show.start <- TTL$start[i] - PREPULSE.PERIOD *SAMPLING.FREQUENCY  
show.stop  <- TTL$stop[i]  + POSTPULSE.PERIOD*SAMPLING.FREQUENCY  
interval.range <- (subsample$Time >= show.start) &  
                  (subsample$Time <= show.stop)
```

```
interval.Time <- subsample$Time[interval.range]  
interval.Value <- subsample$Value[interval.range]
```

Analyzing ABF Files

Finding features and areas in Value channel

Peak and Full-Width at Half Max (FWHM)



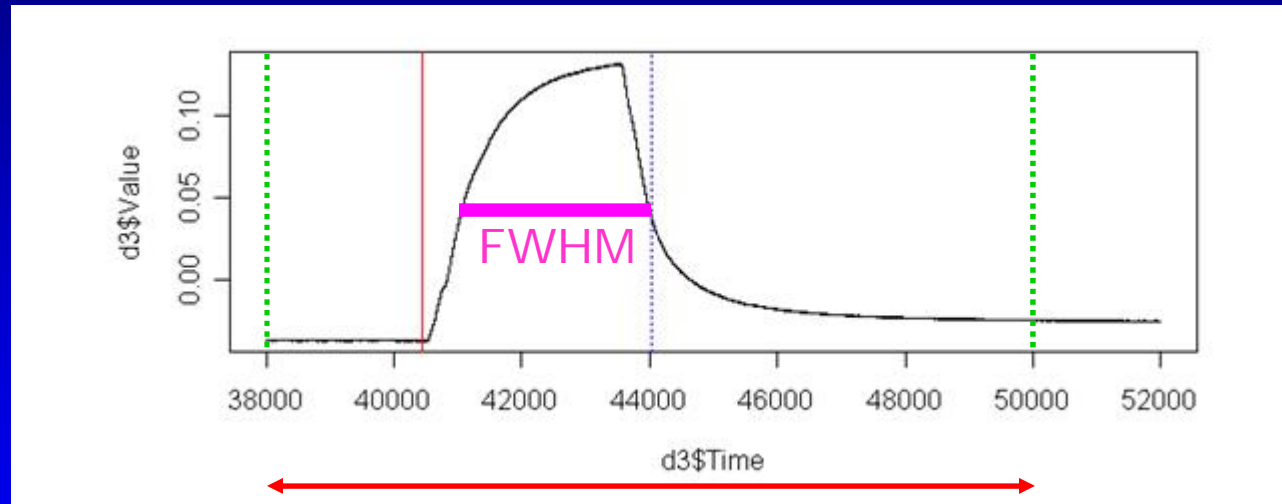
```
# Compute FWHM
Value.max <- max(interval.Value)
Value.min <- min(interval.Value)

Value.halfmax <- (Value.max + Value.min) / 2
# Pick first if more than one
Index.max <- which(interval.Value == Value.max)[1]
```


Analyzing ABF Files

Finding features and areas in Value channel

Full-Width at Half Max (FWHM)

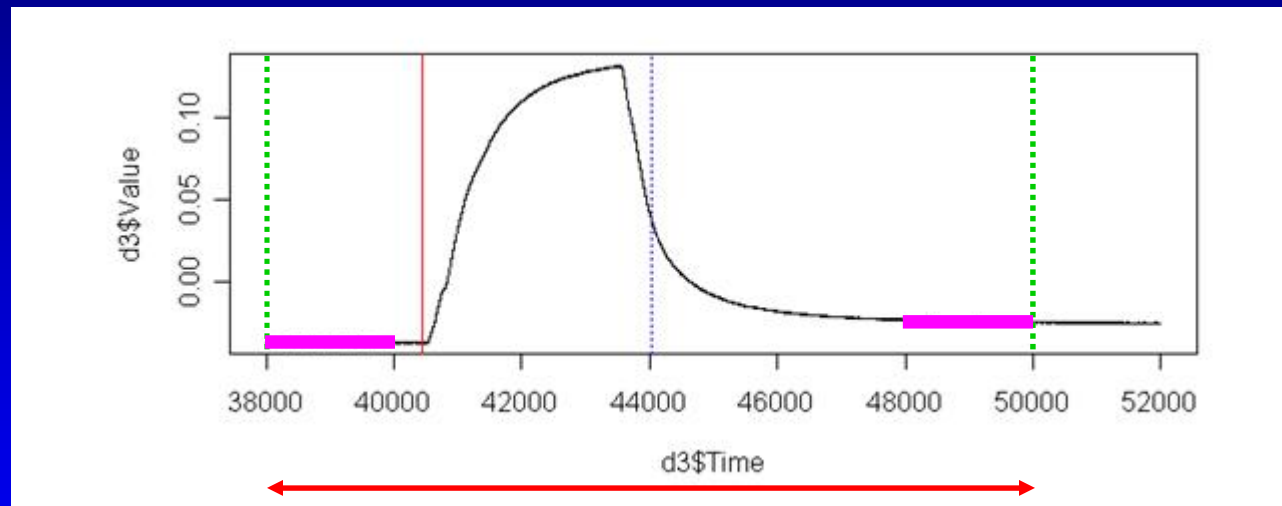


```
suppressWarnings(  
  side.left  <- approx(interval.Value[1:Index.max],  
                        interval.Time[1:Index.max], Value.halfmax))  
  
suppressWarnings(  
  side.right <- approx(interval.Value[Index.max:length(interval.Value)],  
                        interval.Time[Index.max:length(interval.Value)],  
                        Value.halfmax))  
  
FWHM <- side.right$y - side.left$y
```

Analyzing ABF Files

Finding features and areas in Value channel

Left and Right "Plateaus"

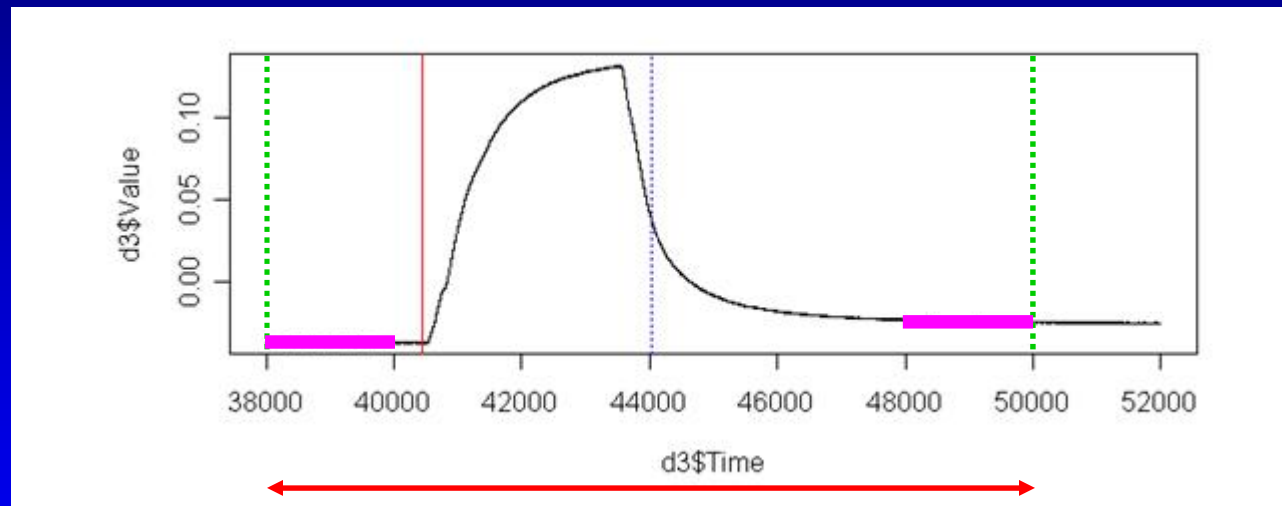


```
plateau.points <- 1:(2*SAMPLING.FREQUENCY %% SUBSAMPLE.FACTOR)
plateau.left   <- median(interval.Value[plateau.points])
plateau.right  <- median(interval.Value[length(interval.Value) -
                                         plateau.points + 1])
plateau.delta  <- plateau.right - plateau.left
plateau.max    <- max(plateau.left, plateau.right)
```

Analyzing ABF Files

Finding features and areas in Value channel

Left and Right "Plateaus"



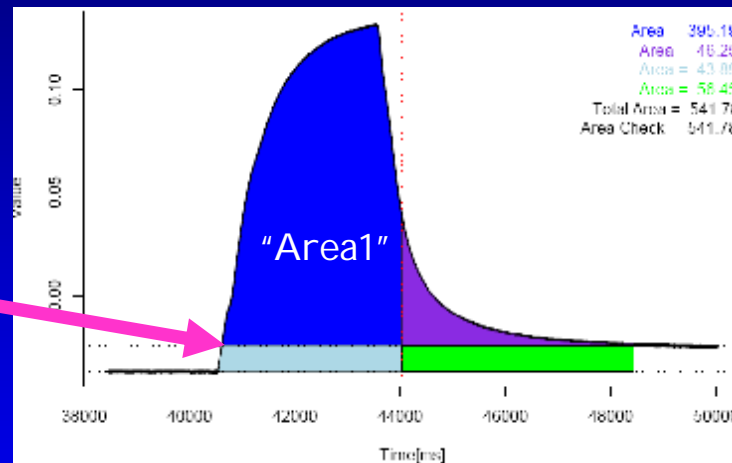
```
plateau.points <- 1:(2*SAMPLING.FREQUENCY %/% SUBSAMPLE.FACTOR)
plateau.left  <- median(interval.Value[plateau.points])
plateau.right <- median(interval.Value[length(interval.Value) -
                                     plateau.points + 1])
plateau.delta <- plateau.right - plateau.left
plateau.max   <- max(plateau.left, plateau.right)
```

Analyzing ABF Files

Finding features and areas in Value channel

Areas

"corner"



```
### Area 1
response.Range <- (interval.Time >= TTL$start[i]) &
  (interval.Time <= TTL$stop[i])
response.Time <- interval.Time[response.Range]
response.Value <- interval.Value[response.Range]

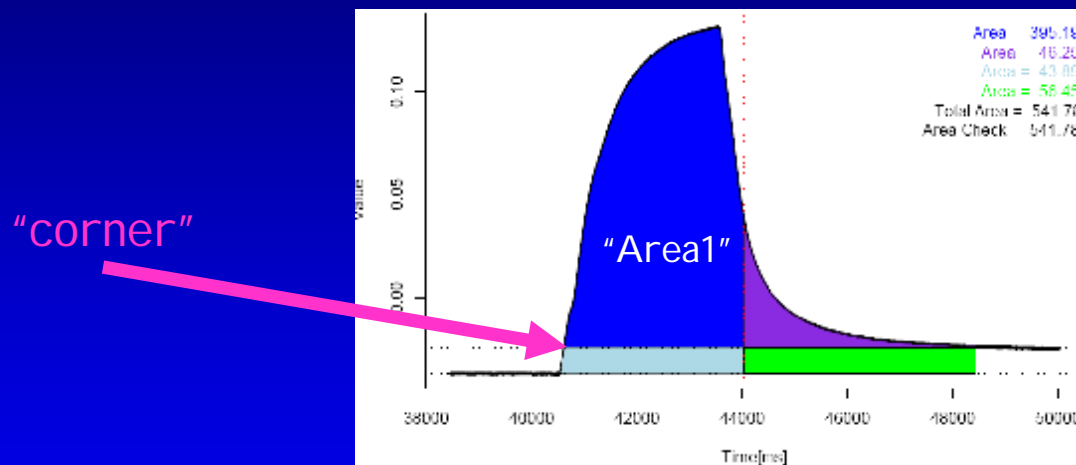
first.above <- which( response.Value > plateau.right)[1]

corner <- approx(response.Value[(first.above-1):first.above],
  response.Time[(first.above-1):first.above],
  plateau.right)
```

Analyzing ABF Files

Finding features and areas in Value channel

Areas



```
response.Subrange <- (response.Time >= corner$y) & (response.Time <= TTL$stop[i])  
polygon1.x <- c(response.Time[response.Subrange], TTL$stop[i], corner$y)  
polygon1.y <- c(response.Value[response.Subrange], plateau.right, plateau.right)  
polygon(polygon1.x, polygon1.y, col="blue")
```

Analyzing ABF Files

Reporting results in charts

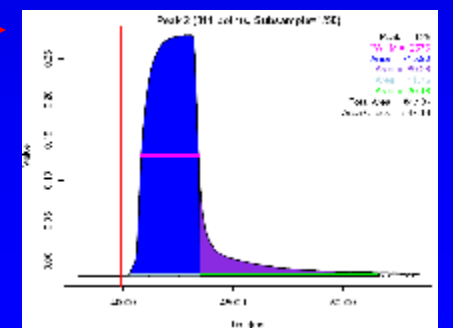
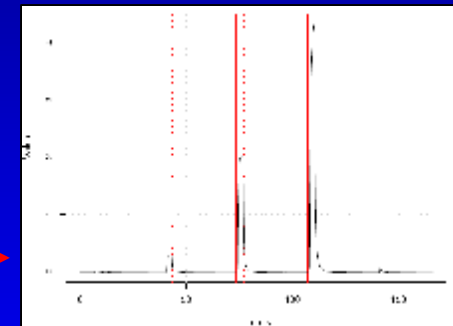
```
process.directory <- function(abf.folder)
{
  abflist <- dir(path=abf.folder, pattern="\\.abf$")
  basefile <- substr(abflist,1,nchar(abflist)-3)
  pdflist <- paste(basefile, "pdf", sep="")
  . . .

  for (file.index in 1:length(abflist))
  {
    abffile <- file.path(abf.folder, abflist[file.index])
    raw <- read.abf.file(abffile)
    TTL <- process.TTL.data(raw)

    subsample <- subsample.raw.data(raw, SUBSAMPLE.FACTOR)

    pdf(file.path(abf.folder, pdflist[file.index]), width=8, height=10)
    par(oma=c(2,0,3,0))
    plot.subsample(subsample, TTL)
    plot.header.and.footer(abffile)

    for (i in 1:length(TTL$start))
    {
      peak.results <- plot.peak(abffile, i, subsample, TTL)
      plot.header.and.footer(abffile)
    }
  }
  dev.off()
}
```



Reporting results in file

```
process.directory <- function(abf.folder)
{
  abflist <- dir(path=abf.folder, pattern="\\.abf$")
  basefile <- substr(abflist,1,nchar(abflist)-3)
  pdflist <- paste(basefile, "pdf", sep="")

  All.Results <- NULL

  for (file.index in 1:length(abflist))
  {
    abffile <- file.path(abf.folder, abflist[file.index])
    cat( format(Sys.time(), "%Y-%m-%d %H:%M:%S"), " Reading", abffile, "\n")
    flush.console()
    raw <- read.abf.file(abffile)
    TTL <- process.TTL.data(raw)

    subsample <- subsample.raw.data(raw, SUBSAMPLE.FACTOR)

    pdf(file.path(abf.folder, pdflist[file.index]), width=8, height=10)
    par(oma=c(2,0,3,0)) # Leave room for footer
    plot.subsample(subsample, TTL)
    plot.header.and.footer(abffile)

    for (i in 1:length(TTL$start))
    {
      peak.results <- plot.peak(abffile, i, subsample, TTL)
      plot.header.and.footer(abffile)

      All.Results <- rbind(All.Results, peak.results)
    }

    dev.off()
  }
  cat( format(Sys.time(), "%Y-%m-%d %H:%M:%S"), "\n")

  write.csv(All.Results, file=file.path(abf.folder, "PeakSummary.csv"), row.names=FALSE)
}
```

Analyzing ABF Files

Reporting results in file

FilePeakSummary.csv										
	A	B	C	D	E	F	G	H	I	J
1	file	peak.index	peak	FWHM	area1	area2	area3	area4	area.total	area.check
2	071225A-AA-25cm-1s-clamp9.abf	1	0.0	2541.2	3.0	1.1	2.4	2.5	9.0	9.0
3	071225A-AA-25cm-1s-clamp9.abf	2	0.3	717.9	188.6	42.9	9.3	34.7	275.6	276.0
4	071225A-AA-25cm-1s-clamp9.abf	3	1.9	769.0	1420.8	213.8	15.0	52.0	1701.6	1701.6
5	071225A-AA-25cm-1s-clamp9.abf	4	2.5	674.6	1639.8	426.5	13.1	46.1	2125.5	2125.6
6	071225A-AA-25cm-1s-clamp9.abf	5	0.2	444.6	87.0	12.7	3.9	12.1	115.6	115.6
7	071225A-AA-25cm-1s.abf	1	0.0	2541.2	3.0	1.1	2.4	2.5	9.0	9.0
8	071225A-AA-25cm-1s.abf	2	0.3	717.9	188.6	42.9	9.3	34.7	275.6	276.0
9	071225A-AA-25cm-1s.abf	3	1.9	769.0	1420.8	213.8	15.0	52.0	1701.6	1701.6
10	071225A-AA-25cm-1s.abf	4	2.5	674.6	1639.8	426.5	13.1	46.1	2125.5	2125.6
11	071225A-AA-25cm-1s.abf	5	0.2	444.6	87.0	12.7	3.9	12.1	115.6	115.6

Several improvements in charts and output file are under consideration.

Take Home: "R" and Analysis Tips

- `system`
- processing directory of files
 - `choose.dir`
 - `dir`
 - `file.path`
 - `rbind` (form composite data.frame)
 - `write.csv`
- cleanup noise in data: `threshold`, `median`
- subsampling data
- area computations
- `approx` to interpolate
- `suppressWarnings`
- FWHM (full width at half max)
- `polygon`

Acknowledgments

Yu Lab

- Nirjal Sapkota
(now at North Carolina State University)
- Limei Ma
- Ron Yu