

# Redox Walkthrough

*Julian Stanley*

*September 27, 2018*

## E vs OxD

```
# Define the Nernst function in terms of oxD
E_OXD <- function(e0, oxD) {
  return(e0 - 12.71 * log((1-oxD)/oxD))
}

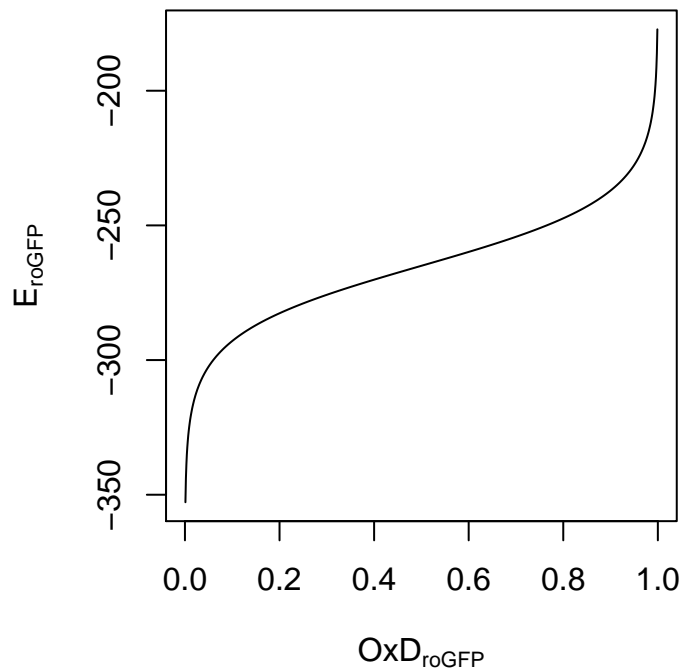
# Generate oxD values between 0 and 1
oxD <- seq(0, 1, by = 0.001)
magOXD <- length(oxD)

# Generate values of E
# Assume: E0 = -265
E = E_OXD(rep(-265, each = magOXD), oxD)

# Set shape and plot
par(pty = 's')

plot(oxD, E, type = 'l', main = "Cell potential as a \n function of fraction oxidized",
      ylab = expression('E'['roGFP']), xlab = expression('OxD'['roGFP']))
```

## Cell potential as a function of fraction oxidized



### OxD(Rmax) vs R

```
# Define the fraction oxidized
OxD <- function(R, Rmin, Rmax, delta) {
  return (
    (R - Rmin)/((R - Rmin) + (delta*(Rmax - R)))
  )
}

# Get some colors
require(RColorBrewer)

## Loading required package: RColorBrewer
## Warning: package 'RColorBrewer' was built under R version 3.4.1
colors <- brewer.pal(n = 8, 'Dark2')

# Define constants and Rmax
# Assume: Minimum ratio of 0.667
# Assume: Delta of 0.171
Rmin <- 0.667
Rmax_values <- c(2, 4, 6, 8, 10, 12)
delta <- 0.171
```

```

# Generate initial values of R
R <- seq(Rmin, Rmax_values[1], by = 0.001)
magR <- length(R)

# Generate initial values of oxD
yOx = OXD(R, rep(Rmin, each = magR),
           rep(Rmax_values[1], each = magR),
           rep(delta, each = magR))

# Set margins and size
par(mar=c(5, 5, 5, 8), pty = 's')

# Plot initial value
plot(R, yOx,
     type = 'l', main = "
     Fraction of molecules oxidized at intensity \n by measured ratio",
     ylab = expression('OxD'['roGFP']), xlab = expression('R'['410/470']),
     xlim = c(0, max(Rmax_values)),
     col = colors[1])

# Plot each variable Rmax
for (i in 2:length(Rmax_values)) {
  # Generate temporary values of R
  R <- seq(Rmin, Rmax_values[i], by = 0.001)
  magR <- length(R)

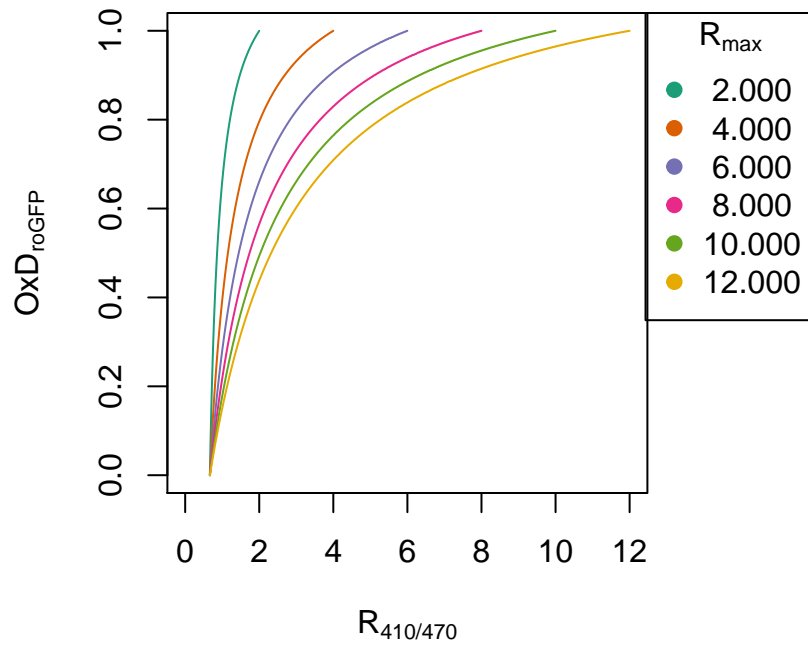
  # Generate temporary values of oxD
  yOx = OXD(R, rep(Rmin, each = magR),
             rep(Rmax_values[i], each = magR),
             rep(delta, each = magR))

  # Plot temporary value
  points(R, yOx, type = 'l', col = colors[i])
}

options(digits = 4)
legend("topright", title = expression(R['max']), inset=c(-0.35,0), xpd=TRUE, as.character(format(Rmax_v

```

## Fraction of molecules oxidized at intensity by measured ratio



```
### OxD(Rmin) vs R
```

```
### OxD(D470) vs R
```

```
### E(Rmax) vs R
```

```
### E(Rmin) vs R
```

```
### E(D470) vs R
```

```
### E(E0) vs R
```

```
### OxD vs R
```

```
### E vs R
```

```
### R -> R' reparameterization
```

```
### dE/dOxD vs OxD
```

```
### dOxD/dR' vs R'
```

```
### dE/dR' vs R'
```

```
### dOxD/dR' (D470) vs R'
```

###  $dE(D470)/dR'$  vs  $R'$

### Empirical Errors

### Background vs Binning

### Image registration/edge effects