

Time-dependent bacterial transcriptional response to CuO nanomaterials differs from that of Cu²⁺ and provides insights into CuO nanomaterial toxicity mechanisms

Joe D. Moore,^{1,2} Astrid Avellan,^{1,2} Clinton W. Noack,¹ Yisong (Alex) Guo,³
Gregory V. Lowry,^{1,2} Kelvin B. Gregory^{1,2*}

¹Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

²Center for the Environmental Implications of NanoTechnology (CEINT), Durham, NC 27708, USA

³Chemistry, Carnegie Mellon University, Pittsburgh, PA 15213, USA

*Corresponding author: 5000 Forbes Avenue, 119 Porter Hall, Pittsburgh, PA 15213, USA. Phone: 412-268-9811. Fax: 412-268-7813. Email: kelvin@cmu.edu

Supporting Information consists of 32 pages, 3 table and 14 figures

Supplemental Methods (p. S2 – S3)

Table S1. Components in MMDM (p. S4)

Table S2. Alternative hypotheses for Mann-Whitney testing of gene expression data for comparisons made in this study (p. S4)

Table S3. Intensity means and polydispersity indices (PdI) of CuO NPs over time (p. S4)

Figure S1. Specificity tests of the CuO NP hyperspectral library signature (p. S5)

Figure S2. HSI of all treatments at 30, 60, and 180 min (p. S6)

Figure S3. *E. coli* growth at various CuO NP and Cu²⁺ concentrations (as Cu) and housekeeping gene (*rrsA*) expression (p. S7)

Figure S4. Treated and undosed *E. coli* viability over time (p. S7)

Figure S5. Gene expression of *btuE* in response to Cu exposure (p. S8)

Figure S6. Gene expression of *recA* in response to CuO NPs (p. S8)

Figure S7. Gene expression of *fabA* in response to Cu exposure (p. S9)

Figure S8. Gene expression of *rpoE* in response to Cu exposure (p. S9)

Figure S9. Gene expression of *spy* in response to Cu exposure (p. S10)

Figure S10. Gene expression of four ROS-responsive genes in response to Cu exposure (p. S11)

Figure S11. X-band EPR spectra of various DMPO treated samples (p. S12)

Figure S12. Raw X-band EPR spectra of various DMPO treated samples without the removal of the Mn²⁺ ion EPR signal (p. S13)

Figure S13. Gene expression of Cu-responsive genes and *sufA* in response to Cu exposure (p. S14)

Figure S14. Induction of induced versus non-induced genes (p. S15)

Works Cited (p. S16)

Supplemental statistical and computational methods (p. S17-S32)

Supplemental Methods

RNA extraction protocol in detail. Immediately following sampling, 2 mL RNAprotect was added to 1 mL of sample, vortexed on high, and allowed to incubate at room temperature for 10 min. The sample was then centrifuged for 10 min at 5000g. TE buffer with 1 mg/L lysozyme was added to the pellet, vortexed, and incubated for 10 min with shaking. Freshly made 2-mercaptoethanol-RLT buffer was added and vortexed, then 100% ethanol was added and mixed by pipetting before being transferred to a spin column and centrifuged. RW1 buffer was added and centrifuged through the spin column before on-column DNase 1 treatment according to the manufacturer's protocol. Freshly made RPE-ethanol was added to the spin column twice. Spin columns were transferred to a clean 1.5 mL tube and allowed to air dry for 5 min. RNA was eluted into 60 μ L RNase-free 45°C water that was divided into two 30 μ L additions with 3 min incubation before centrifuging.

Growth inhibition assays. Inhibition of *E. coli* growth by varying concentrations of CuO ENMs and Cu²⁺ was measured using 96-well plates and a microplate reader (SpectraMax, Molecular Devices, Sunnyvale, CA). Approximately 10⁶ colony forming units (CFU) per mL were added to a 96-well plate containing serially diluted CuO NPs or Cu²⁺ in the growth medium. The plate was shaken at 160 rpm at 30°C, and OD600 was measured at 0 hr and 24 hr. Growth was determined by subtracting the 0 hr reading from the 24 hr reading. Abiotic and undosed biotic controls were included alongside CuO NP and Cu²⁺ treatments, and eight wells were used per treatment or control condition. Stocks of Cu²⁺ were made from Cu(NO₃)₂ in ultrapure water. ICP-MS analysis (as described above) was used to ensure that nominal Cu concentrations agreed with analytical concentrations.

Both CuO NPs and Cu²⁺ demonstrated the expected dose response, with higher Cu concentrations leading to lower bacterial growth at 24 hr (**Figure S2**). Cu²⁺ inhibited growth at ~1 ppm, while CuO NPs allowed growth out to ~20 mg/L Cu. Previous studies have observed greater bacterial susceptibility to dissolved Cu than Cu ENMs.^{1,2} The CuO NP data appear to show an increase in *E. coli* growth at low concentrations. This increase has been attributed to hormesis for exposures to Ag NPs.³ However, follow up plating experiments showed that CFUs did not increase at the same low CuO NP concentrations (data not shown), suggesting that the low CuO NP concentration optical density increases were an artefact of the measurements.

Discussion of CuO NP dosing concentration. We chose to dose at 100 mg/L (as Cu) CuO NPs to ensure that the NPs imposed a stress on the *E. coli*. This concentration has also been used in a previous study of CuO NP bacterial toxicity.⁴ Inhibition assays with CuO NPs showed that 100 mg/L inhibited the growth of 10⁶ *E. coli* CFU/mL. At the higher bacterial concentration needed for the transcriptional assays (5 x 10⁷ CFU/mL) 100 mg/L CuO NPs provided a sublethal stress based on the observed increase in *rtsA* gene expression over time.

Viability experiments. *E. coli* was grown and treated with Cu as described in the main text for HSI and transcriptional assays. All treatments were done in duplicate, including an undosed control. After 30, 60, 100, 140, and 180 min of exposure, samples were vortexed to homogenize, and 100 μ L samples of each sample were serially diluted. 50 μ L of serially diluted samples were added to LB agar plates, again in duplicate. Plates were incubated at 37°C, and CFUs were counted 24 hr later. Means and standard deviations were calculated at each time point.

Oxidative stress probe protocol in detail. Cells were grown to early log phase growth as with the transcriptional response experiments. At OD ~ 0.2, cells were pelleted by centrifuging for 15 min at 8000g. While the cells were centrifuging, a 10 mM stock of CM-H2DCFDA (Molecular Probes, Life

Technologies) was made by adding DMSO to a single CM-H2DCFDA tube in the dark. This 10 mM stock was diluted to 10 μ M in pH 7.2 phosphate buffered saline (PBS). The cells were resuspended in the PBS containing the probe and allowed to incubate with gentle shaking for 30 min to allow for the probe to penetrate into the cells. The cells were then centrifuged and suspended back in an appropriate volume of growth medium for 5×10^7 CFU/mL. Treatments were added immediately. At 10, 60, and 180 min, samples were vortexed and 200 μ L were withdrawn from each sample in triplicate and added to a 96 well plate. Fluorescence emitted at 517 nm was measured immediately on a microplate reader after excitation at 494 nm.

Electron paramagnetic resonance (EPR) methods in detail. EPR was conducted using 5,5-dimethyl-1-pyrroline-N-oxide (DMPO) (Cayman Chemical, Ann Arbor, MI) as a radical spin trap. After 60 and 180 min Cu-exposed and undosed bacteria were incubated with 20 mM DMPO for 15 min at room temperature before being frozen with liquid N₂, where they were stored until EPR measurements were taken. Positive controls in H₂O and MMDM were generated by reacting 0.1 mM Fe²⁺, 1 mM H₂O₂ and 20 mM DMPO in a glovebox (MBraun, Garching, Germany) filled with N₂ gas (O₂ < 0.5 ppm) for 5 min at room temperature, then were frozen in liquid N₂. Negative controls were generated by incubating 20 mM DMPO with abiotic MMDM, *E. coli* in MMDM, Cu²⁺ in MMDM, and CuO NPs in MMDM. X-band EPR spectra were measured on a Bruker Elexsys E-500 spectrometer (microwave frequency = 9.64 GHz, Billerica, MA) equipped with an Oxford ESP-910 cryostat (Abingdon, United Kingdom). All spectra were measured at 73 K with 0.3 mT modulation amplitude, 100 kHz modulation frequency. EPR spin quantification was done by comparing the double integration of an EPR signal with unknown concentration with the double integration of the EPR signal from a known concentration standard measured under the same measurement conditions. The spin standard used in this study is 1.2 mM Cu(II)-EDTA solution. EPR spectra were processed and plotted using SpinCount software.⁵

EPR spectral interpretation. The DMPO treated samples included in this study exhibited three types of EPR-detectable signals. Two EPR signals were observed in all samples prepared in MMDM. The signal spreading across a wide magnetic field range (from 320 mT to 370 mT) is characteristic of Mn²⁺ (**Figure S11**). The signal exhibiting a sharp g = 2 resonance belongs to certain organic radical with unknown origin (**Figures S10g and S10h**). The spin concentration quantification suggested that the Mn²⁺ ion signal was < 20 μ M, and the organic radical signal was << 1 μ M. In all Cu²⁺ or CuO NP dosed samples, a third EPR signal was observed with various spin concentrations (**Figure S10 a-f**), which was centered at g ~ 2 with a four-line splitting pattern. This EPR signal was reproduced (**Figures S10i and S10j**) in two positive control samples with the presence of DMPO under Fenton chemistry reaction conditions (see the experiment section for the details of sample preparation), suggesting that this signal belongs to a DMPO radical adduct formed in the presence of \cdot OH. The spectral analysis suggested that this four-line splitting EPR signal can be simulated by including a ¹⁴N hyperfine coupling and ¹H hyperfine coupling interactions with hyperfine coupling constants of A(¹⁴N) = 60 MHz and A(¹H) = 65 MHz (**Figure S10j**). Although these parameters do not resemble the ones from the DMPO-OH radical adduct (A(¹⁴N) = A(¹H) = 40 MHz) that are expected to be generated from the spin trap experiment, these parameters are comparable to the ones from a DMPO radical adduct with carbon centered radicals (A(¹⁴N) = 45 MHz and A(¹H) = 65 MHz).⁶ DMSO was used as the solvent to prepare our DMPO stock solution. It has been reported that \cdot OH will activate DMSO to form a carbon centered radical (\cdot CH₃), and further form the DMPO-CH₃ radical adduct.⁶ Therefore, due to the presence of DMSO during our spin trap experiment, the DMPO-CH₃ radical adduct is observed in our systems.

Table S1. Components in MMDM. Ionic strength was estimated to be 0.485 using Visual Minteq (version 3.1) equilibrium modeling software.

mM with pH 6.5 MDM + supplements	
K	103
PO4	94
NH4	15
SO4	171
Na	174
Mg	123
Mn	30
Cl	192
Fe	3.6
NO3	6.9
Ca	9.1
Zn	6.2
MoO4	0.41
SeO3	0.06
WO4	0.30
Ni	0.84

Table S2. Alternative hypotheses for Mann-Whitney testing of gene expression data for comparisons made in this study. H_1 refers to the value of the location shift, μ , between sample distributions of gene (x) expression at time (t) between treatments (y or *undosed*).

Statistical comparison	Biological question	H_1
Treatment vs. control, <i>rrsA</i> expression only	Is bacterial growth negatively impacted by Cu exposures?	$\mu_{rrsA,t}^{y-undosed} < 0$
Treatment vs. control, all other genes	Does Cu exposure lead to differential expression of stress-response genes?	$\mu_{x,t}^{y-undosed} > 0$
Treatment vs. treatment, all genes	Does the mode of Cu exposure lead to differential expression of stress-response genes?	$\mu_{x,t}^{y_1-y_2} \neq 0$

Table S3. Intensity means and polydispersity indices (PdI) of CuO NPs over time.

Time [min]	Intensity mean [nm]	PdI
0	638 ± 150	0.356 ± 0.049
10	766 ± 81	0.402 ± 0.011
60	707 ± 160	0.422 ± 0.080
180	n/a	n/a

Library, specificity test

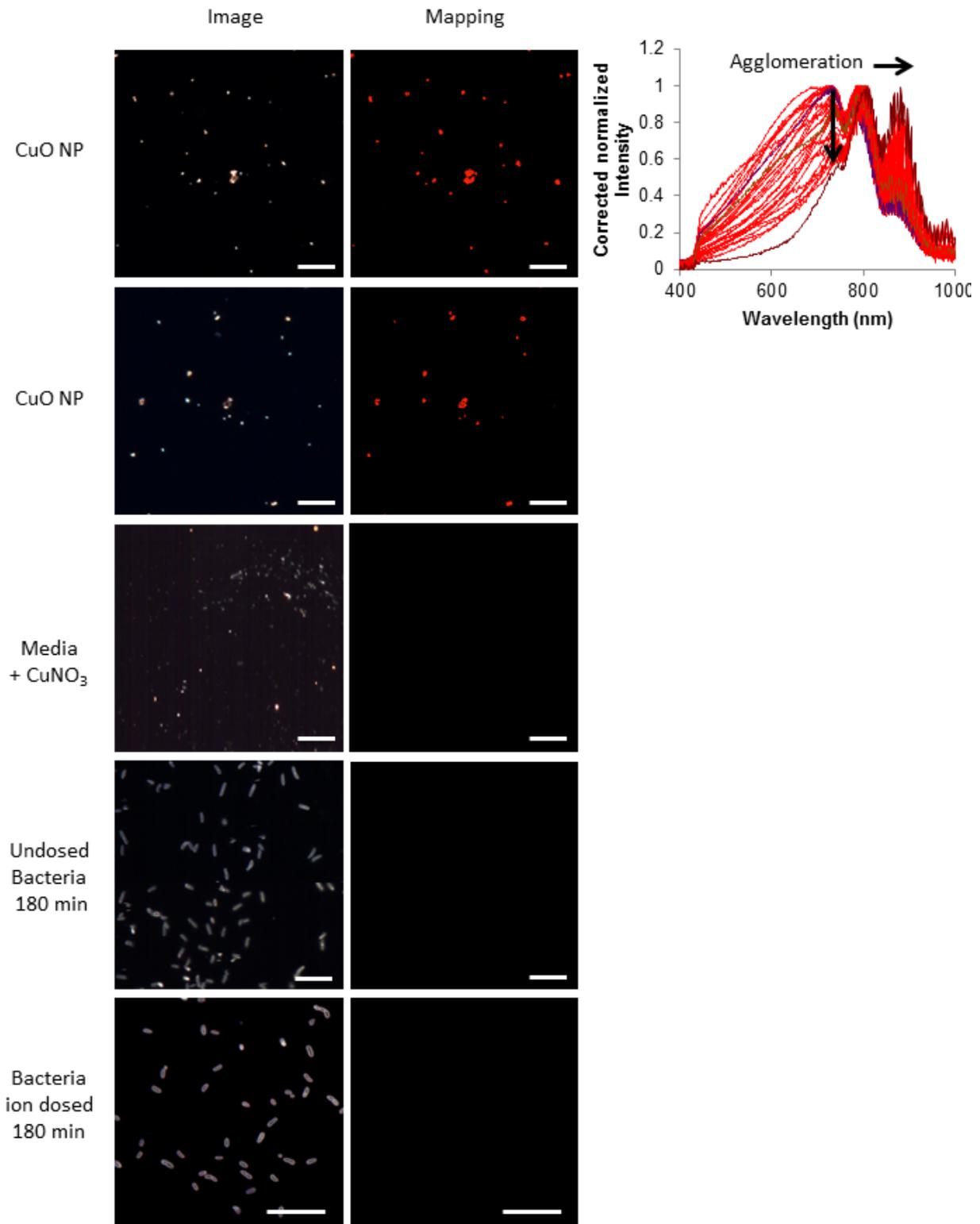


Figure S1. Specificity tests of the CuO NP hyperspectral library signature.

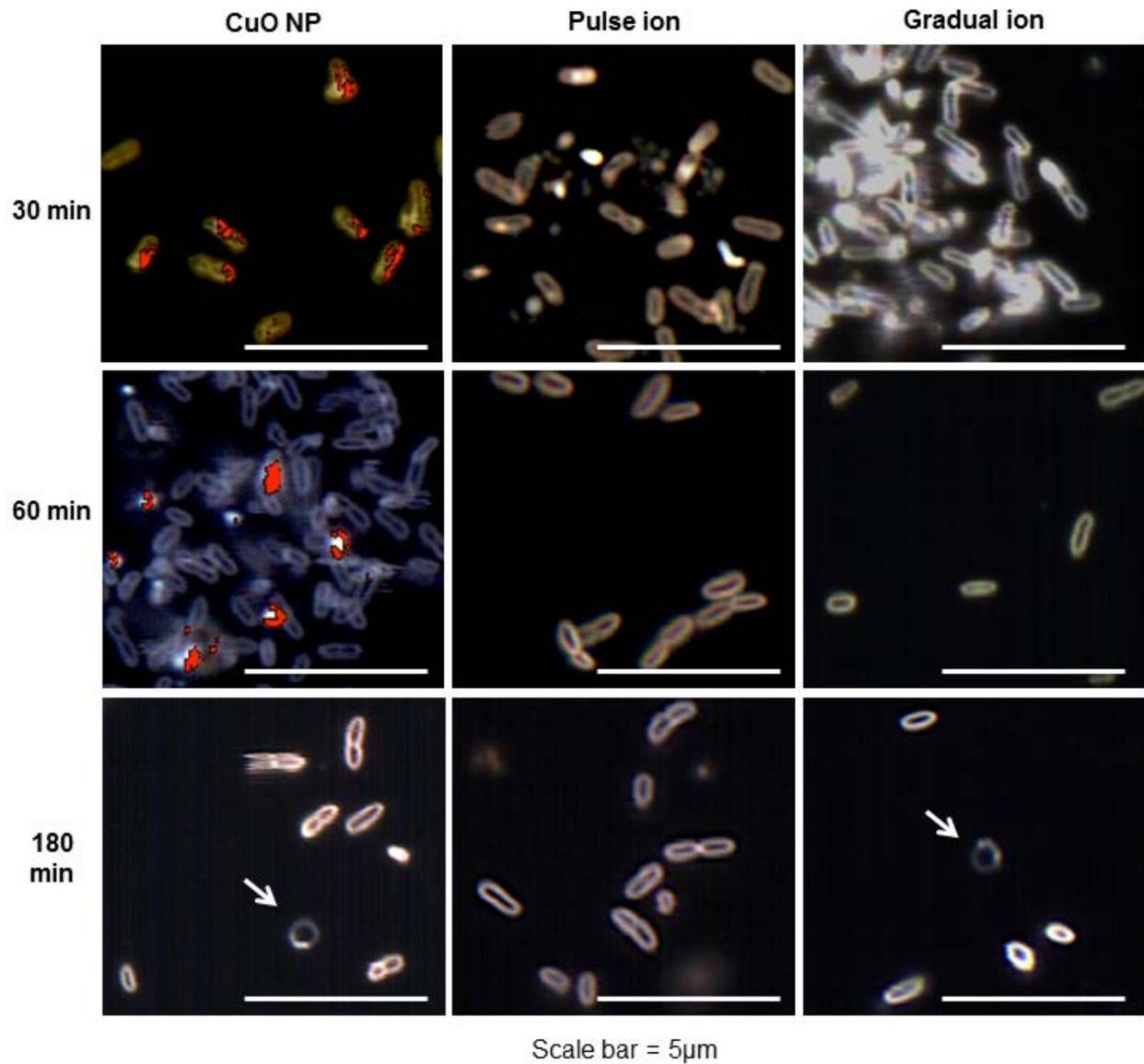


Figure S2. HSI of all treatments at 30, 60, and 180 min. Arrows point to OMVs detected at 180 min with NP and gradual ion treatment.

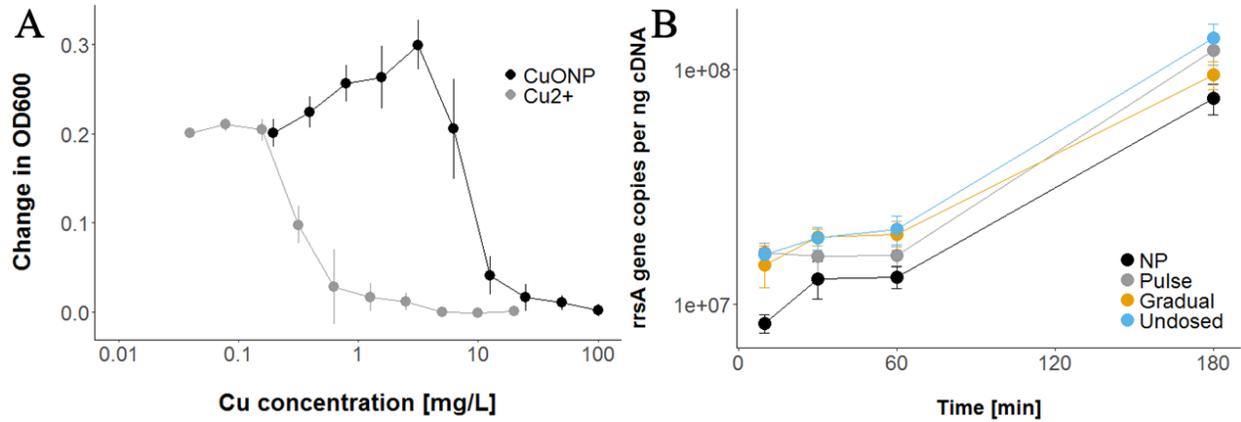


Figure S3. *E. coli* growth at various CuO NP and Cu²⁺ concentrations (as Cu) and housekeeping gene (*rrsA*) expression. (A) Growth was measured by change in optical density between 0 and 24 hr. (B) *rrsA* gene expression was statistically similar across treatments and showed at 180 min the enhanced expression that is characteristic of entering exponential growth.

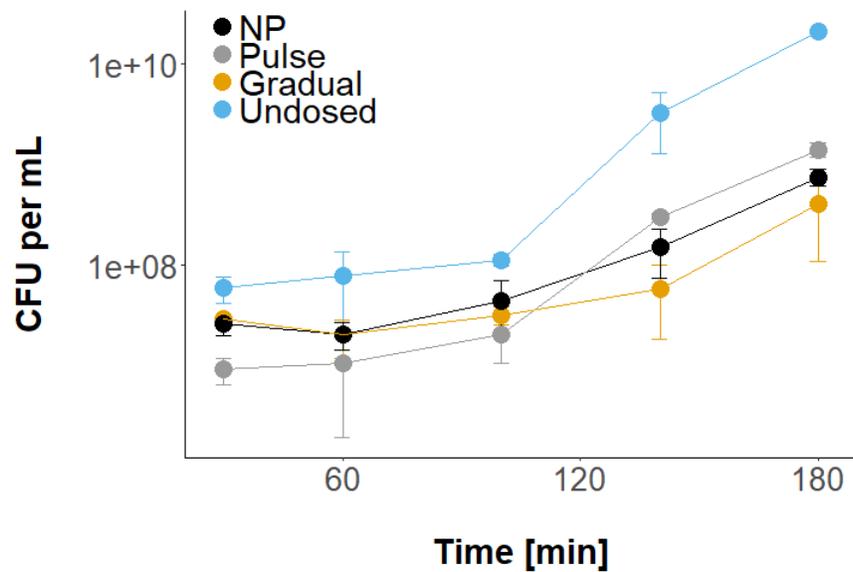


Figure S4. Treated and undosed *E. coli* viability over time. Results of time-resolved CFU experiment for all treatments on a y-axis log scale.

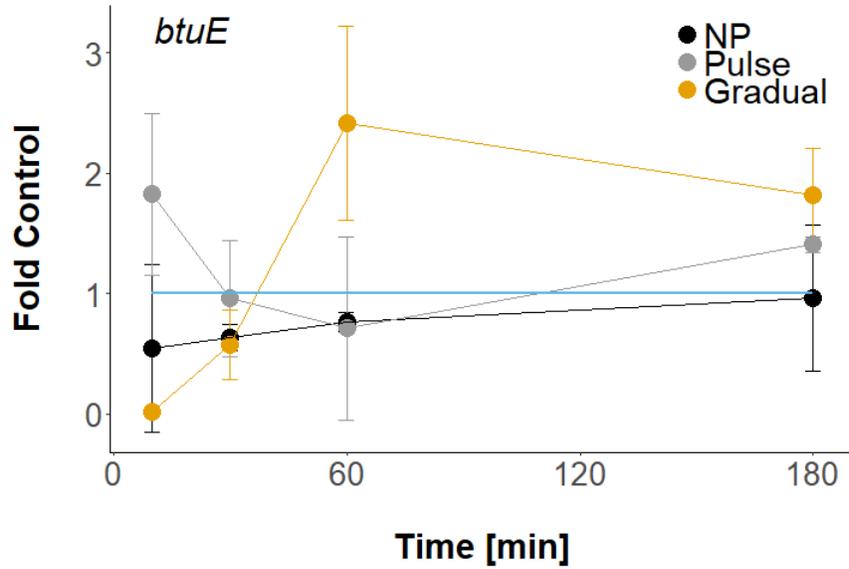


Figure S5. Gene expression of *btuE* in response to Cu exposure. Statistically significant differences were observed at 10 min. The blue horizontal line is fixed at one to represent the expression of the undosed control. Error bars represent one SD of triplicate samples.

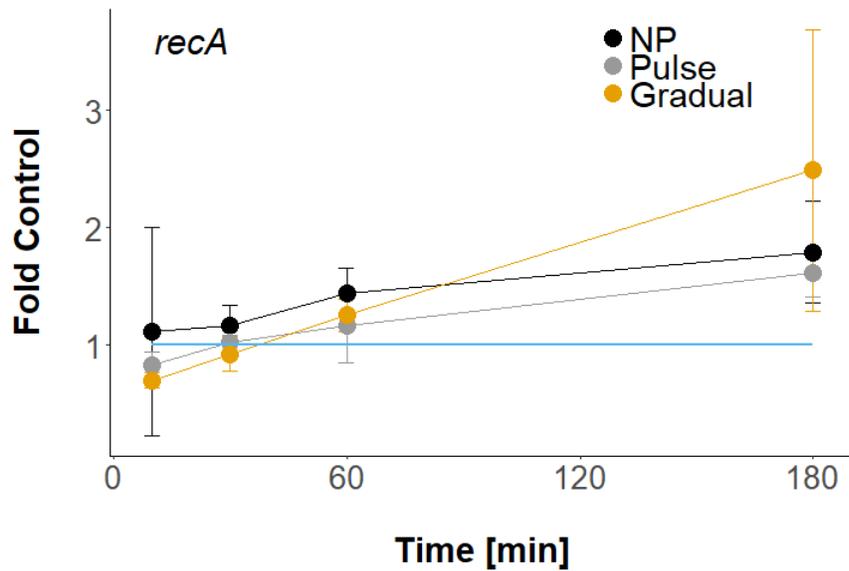


Figure S6. Gene expression of *recA* in response to CuO NPs. The blue horizontal line is fixed at one to represent the expression of the undosed control. Error bars represent one SD of triplicate samples.

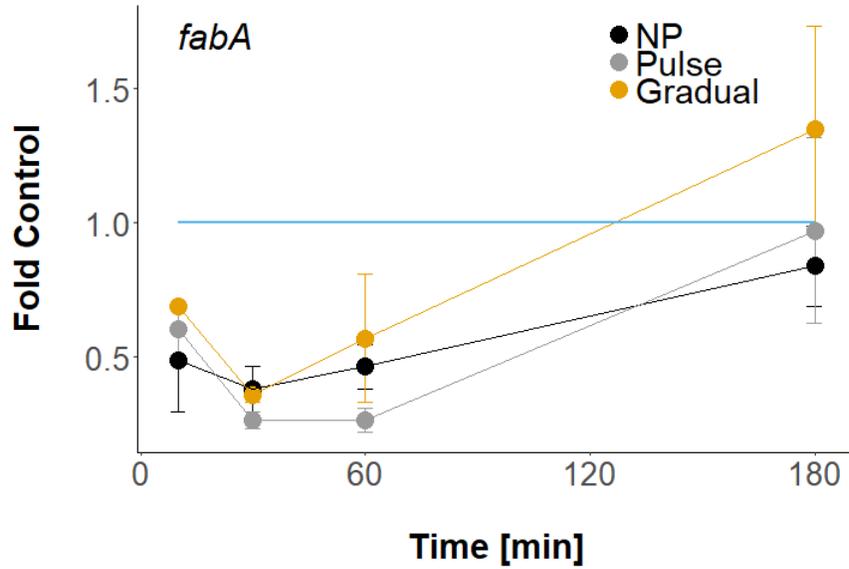


Figure S7. Gene expression of *fabA* in response to Cu exposure. Statistically significant differences were observed at 60 min. The blue horizontal line is fixed at one to represent the expression of the undosed control. Error bars represent one SD of triplicate samples.

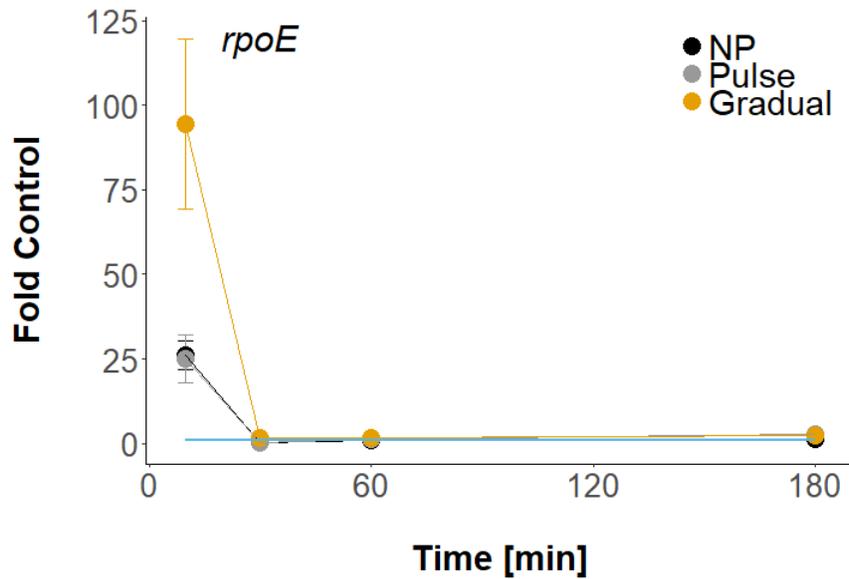


Figure S8. Gene expression of *rpoE* in response to Cu exposure. Statistically significant differences were observed at 30 min. The blue horizontal line is fixed at one to represent the expression of the undosed control. Error bars represent one SD of triplicate samples.

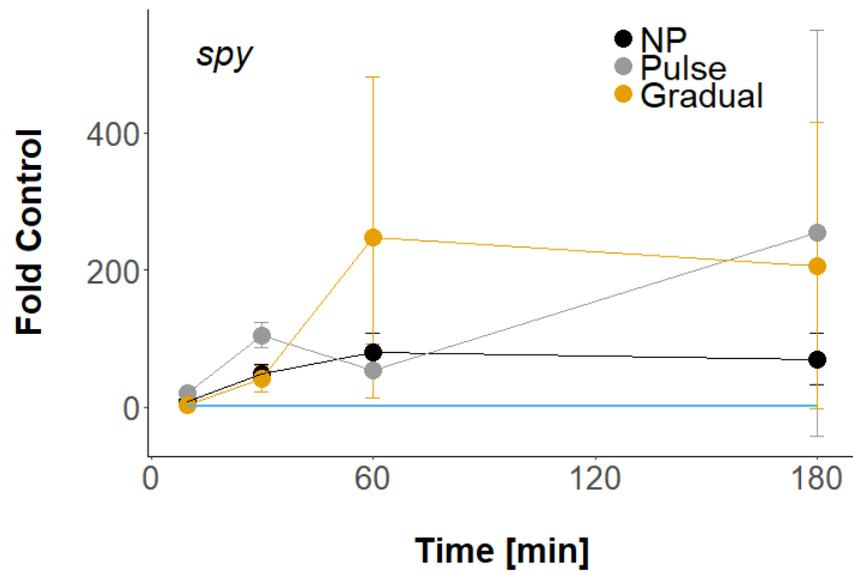


Figure S9. Gene expression of *spy* in response to Cu exposure. Statistically significant differences were observed at 30 min. The blue horizontal line is fixed at one to represent the expression of the undosed control. Error bars represent one SD of triplicate samples.

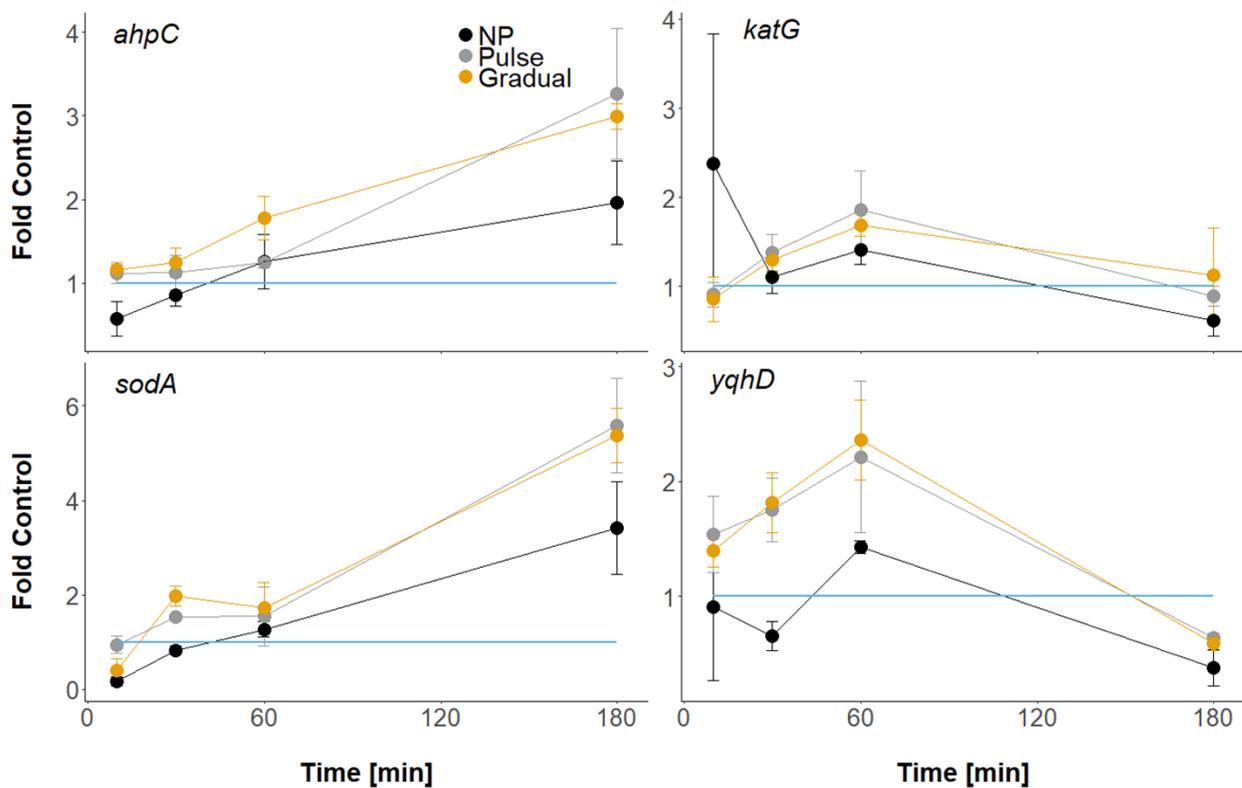


Figure S10. Gene expression of four ROS-responsive genes in response to Cu exposure. Details regarding each gene can be found in the main text. Statistically significant differences were observed for *ahpC* at 10 and 180 min; for *sodA* at 10, 30, and 180 min; and for *yqhD* at 30 and 60 min. The blue horizontal line is fixed at one to represent the expression of the undosed control. Error bars represent one SD of triplicate samples.

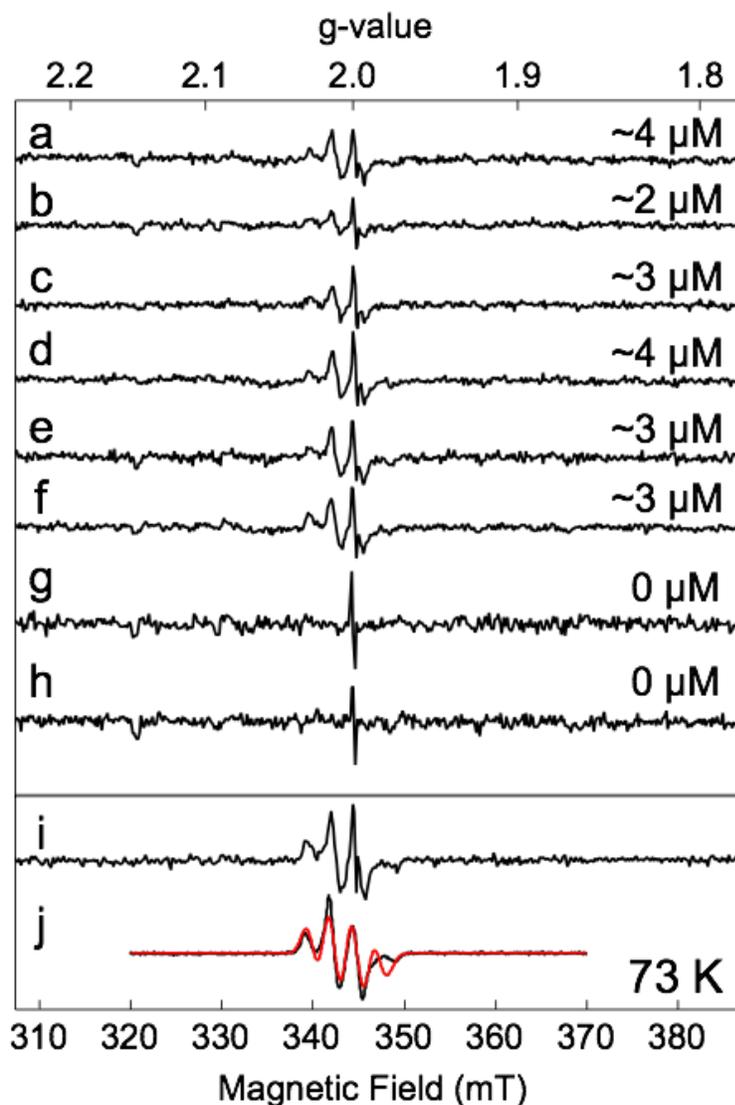


Figure S11. X-band EPR spectra of various DMPO treated samples. (a) Gradual Cu^{2+} -dosed bacteria grown to 180 min; (b) Gradual Cu^{2+} -dosed bacteria at 60 min; (c) pulse Cu^{2+} -dosed bacteria at 180 min; (d) CuO NP-dosed bacteria at 180 min; (e) Cu^{2+} -dosed MMDM; (f) CuO NP-dosed MMDM; (g) undosed bacteria at 180 min; (h) abiotic MMDM only; (i) positive control in MMDM; (j) positive control in H_2O (black) and the spectral simulation (red) using the following parameters: $g = 2.005$, $A(^{14}\text{N}) = 60$ MHz, $A(^1\text{H}) = 65$ MHz. The Mn^{2+} ion signal has been removed for each spectrum except for spectrum j, where no Mn^{2+} ion was present. The spin concentrations of the DMPO radical adduct in spectra (a) – (h) are indicated in the figure. Measurement conditions: microwave frequency, 9.64 GHz; microwave power, 20 μW ; modulation amplitude, 0.3 mT; modulation frequency, 100 kHz; temperature, 73 K.

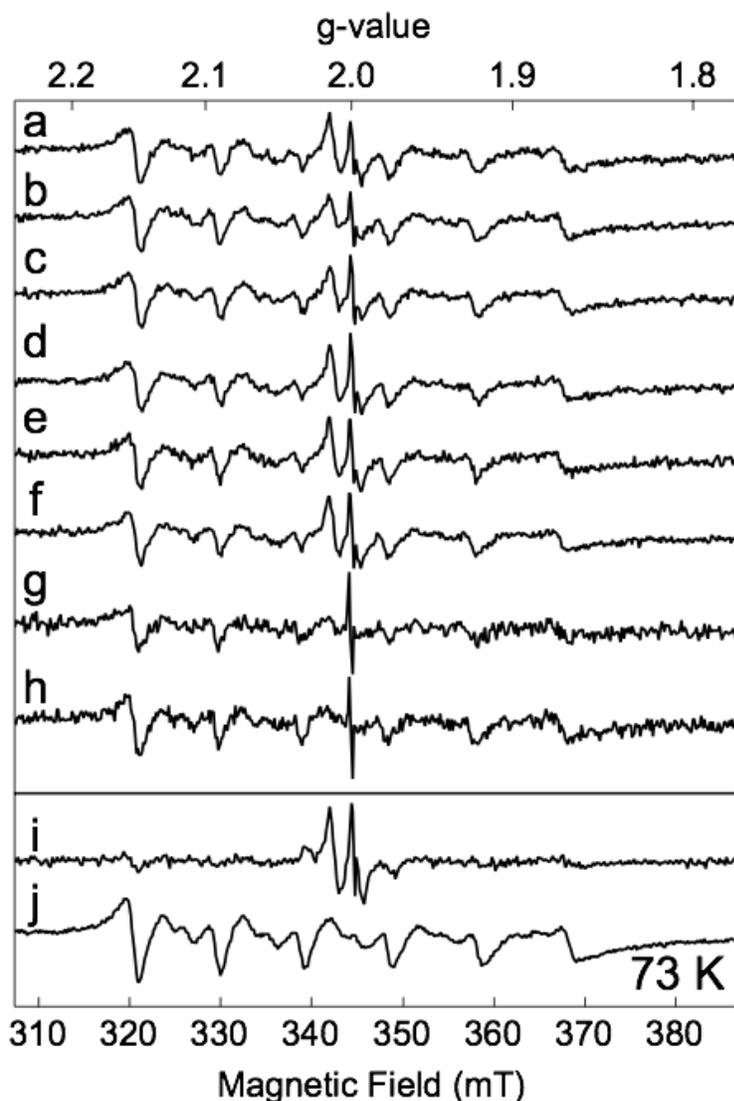


Figure S12. Raw X-band EPR spectra of various DMPO treated samples without the removal of the Mn^{2+} ion EPR signal. (a) Gradual Cu^{2+} -dosed bacteria at 180 min; (b) Gradual Cu^{2+} -dosed bacteria at 60 min; (c) pulse Cu^{2+} -dosed bacteria at 180 min; (d) CuO NP-dosed bacteria at 180 min; (e) Cu^{2+} -dosed MMDM; (f) CuO NP-dosed MMDM; (g) undosed bacteria at 180 min; (h) abiotic MMDM only; (i) positive control in MMDM; (j) the typical Mn^{2+} ion EPR spectrum measured in the MMDM only. Measurement conditions: microwave frequency, 9.64 GHz; microwave power, 20 μW (except for spectrum j, which was measured using 2 mW); modulation amplitude, 0.3 mT; modulation frequency, 100 kHz; temperature, 73 K.

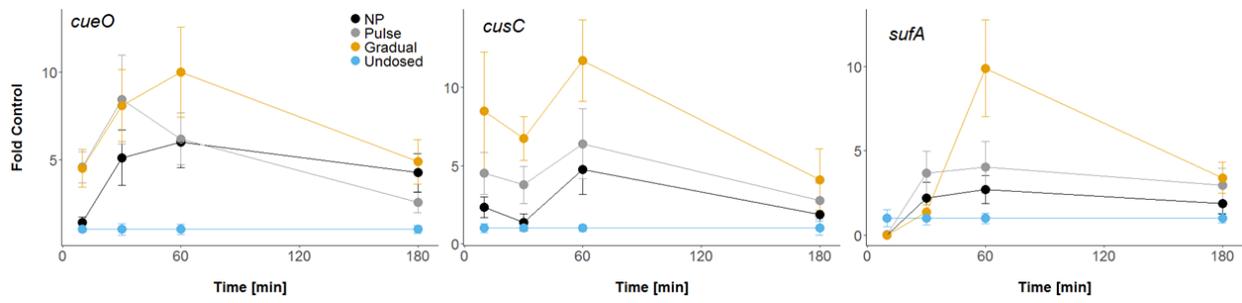


Figure S13. Gene expression of Cu-responsive genes and *sufA* in response to Cu exposure. Details regarding each gene can be found in the main text.

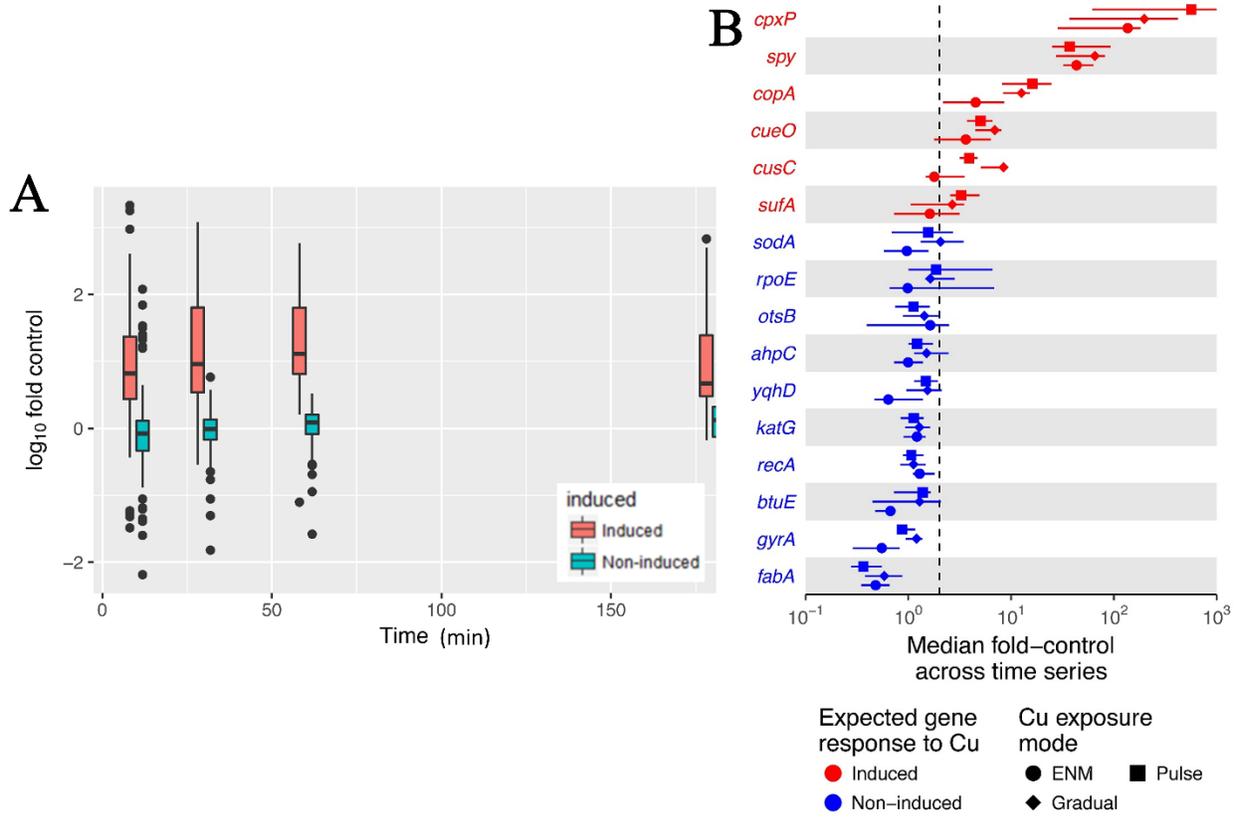


Figure S14. Induction of induced versus non-induced genes. Induced genes (*copA*, *cueO*, *cusC*, *sufA*, *cpxP*, and *spy*) demonstrate induction over the course of the experiment, but especially through 60 min, whereas non-induced genes do not (A). Median fold control over the whole time series distinguishes induced versus non-induced genes as well (B). The dashed line demarcates 2-fold control expression.

Works Cited

- 1 C. Gunawan, W. Y. Teoh, C. P. Marquis and R. Amal, *ACS Nano*, 2011, 7214–7225.
- 2 C. Kaweeteerawat, C. H. Chang, K. R. Roy, R. Liu, R. Li, D. Toso, H. Fischer, A. Ivask, Z. Ji, J. I. Zink, Z. H. Zhou, G. F. Chanfreau, D. Telesca, Y. Cohen, P. A. Holden, A. E. Nel and H. A. Godwin, *ACS Nano*, 2015, **9**, 7215–7225.
- 3 Z. Xiu, Q. Zhang, H. Puppala, V. Colvin and P. Alvarez, *Nano Lett.*, 2012, 10–14.
- 4 G. Applerot, J. Lellouche, A. Lipovsky, Y. Nitzan, R. Lubart, A. Gedanken and E. Banin, *Small*, 2012, **8**, 3326–37.
- 5 D. T. Petasis and M. P. Hendrich, 2015, pp. 171–208.
- 6 G. R. Buettner, *Free Radic. Biol. Med.*, 1987, **3**, 259–303.

Supplemental statistical and computational methods

This section is meant to more thoroughly describe and divulge the computational methodologies utilized in the main text as well as provide additional insight into results where appropriate. Specifically, the text here will walk through the relevant steps in the analysis of gene expression data and oxidative-stress measured by fluorescence.

This section was written in R Markdown v1.5 using RStudio and contains all the relevant code necessary to reproduce the analysis presented in the manuscript.

Guide to computational methods

Analysis dependencies

The code makes extensive use of the packages `plyr`, `broom`, `magrittr`, `dplyr`, `tidyr`, and `ggplot2` (where the latter three packages are loaded as part of the `tidyverse` package) in addition to the functions present in the base R language. Therefore, these packages need to be installed by the R user in order to reproduce this analysis. Aside from plotting commands from `ggplot2` (which will be acknowledged with in-script comments: `# This is a comment`) and the pipe operator (`%>%`; loaded via the `dplyr` namespace, but part of the `magrittr` package), functions from external packages will be indicated by `package_name::function_name`. When these packages are attached *via* the `library()` function (as they are here), this syntax is technically redundant. However, this convention will still be used here to provide attribution to the appropriate source throughout.

```
## Uncomment these lines and execute in R if these packages are not already installed.
# install.packages('plyr')
# install.packages('broom')
# install.packages('tidyverse')
# install.packages('magrittr')

library(plyr)
library(broom)
library(tidyverse)
library(magrittr)
```

Analysis environment

The code given herein runs without errors under the following configuration; no assurances are made for backward or forward compatibility of R versions or versions of the dependencies.

```
## R version 3.4.0 (2017-04-21)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: OS X El Capitan 10.11.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
```

```
##
## other attached packages:
## [1] magrittr_1.5      dplyr_0.7.1      purrr_0.2.2.2    readr_1.1.1
## [5] tidyr_0.6.3       tibble_1.3.3     ggplot2_2.2.1    tidyverse_1.1.1
## [9] broom_0.4.2       plyr_1.8.4
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.11      cellranger_1.1.0 compiler_3.4.0    bindr_0.1
## [5] forcats_0.2.0     tools_3.4.0      digest_0.6.12    lubridate_1.6.0
## [9] jsonlite_1.5      evaluate_0.10.1  nlme_3.1-131     gtable_0.2.0
## [13] lattice_0.20-35   pkgconfig_2.0.1  rlang_0.1.1      psych_1.7.5
## [17] yaml_2.1.14       parallel_3.4.0   haven_1.0.0      bindrcpp_0.2
## [21] xml2_1.1.1        httr_1.2.1       stringr_1.2.0    knitr_1.16
## [25] hms_0.3           rprojroot_1.2    grid_3.4.0       glue_1.1.1
## [29] R6_2.2.2          readxl_1.0.0     foreign_0.8-67   rmarkdown_1.6
## [33] modelr_0.1.0      reshape2_1.4.2   backports_1.1.0  scales_0.4.1
## [37] htmltools_0.3.6   rvest_0.3.2      assertthat_0.2.0 mnormt_1.5-5
## [41] colorspace_1.3-2 stringi_1.1.5     lazyeval_0.2.0   munsell_0.4.3
```

Gene expression data, quality control, and expression normalization

Gene expression data are stored in a CSV file accompanying this manuscript (first few rows shown below). The 16S rRNA gene, *rrsA*, was used as a housekeeping gene against which all other target genes were normalized. *rrsA* has been validated for this function before. The variables in this dataset are: **Treatment** (the mode of Cu exposure), **Time** (the duration of exposure in minutes), **Gene** (the target gene), and **Expression** (the expression of the target gene relative to *rrsA*). For **Treatment** the variable levels are: **Undosed** (the experimental control with no Cu exposure), **ENM** (the nano-Cu dosing), **Pulse** (immediate 100 mg/L dissolved Cu dosing), and **Gradual** (incremental dissolved Cu dosing to match observe ENM dissolution kinetics).

```
# add an identifier of the experimental replicates
gene_expression <- read.csv('gene-expression-time-series-UPDATED.csv') %>%
  dplyr::group_by(Treatment, Time, Gene) %>%
  dplyr::mutate(rep = 1:n()) %>%
  dplyr::ungroup()
dplyr::tbl_df(gene_expression)
```

```
## # A tibble: 799 x 5
##   Treatment Time Gene Expression rep
##   <fctr> <int> <fctr> <dbl> <int>
## 1 ENM 10 rrsA 4390000 1
## 2 ENM 10 rrsA 8000000 2
## 3 ENM 10 rrsA 12500000 3
## 4 Pulse 10 rrsA 16300000 1
## 5 Pulse 10 rrsA 17300000 2
## 6 Pulse 10 rrsA 16300000 3
## 7 Gradual 10 rrsA 14700000 1
## 8 Gradual 10 rrsA 14800000 2
## 9 Undosed 10 rrsA 14400000 1
## 10 Undosed 10 rrsA 17000000 2
## # ... with 789 more rows
```

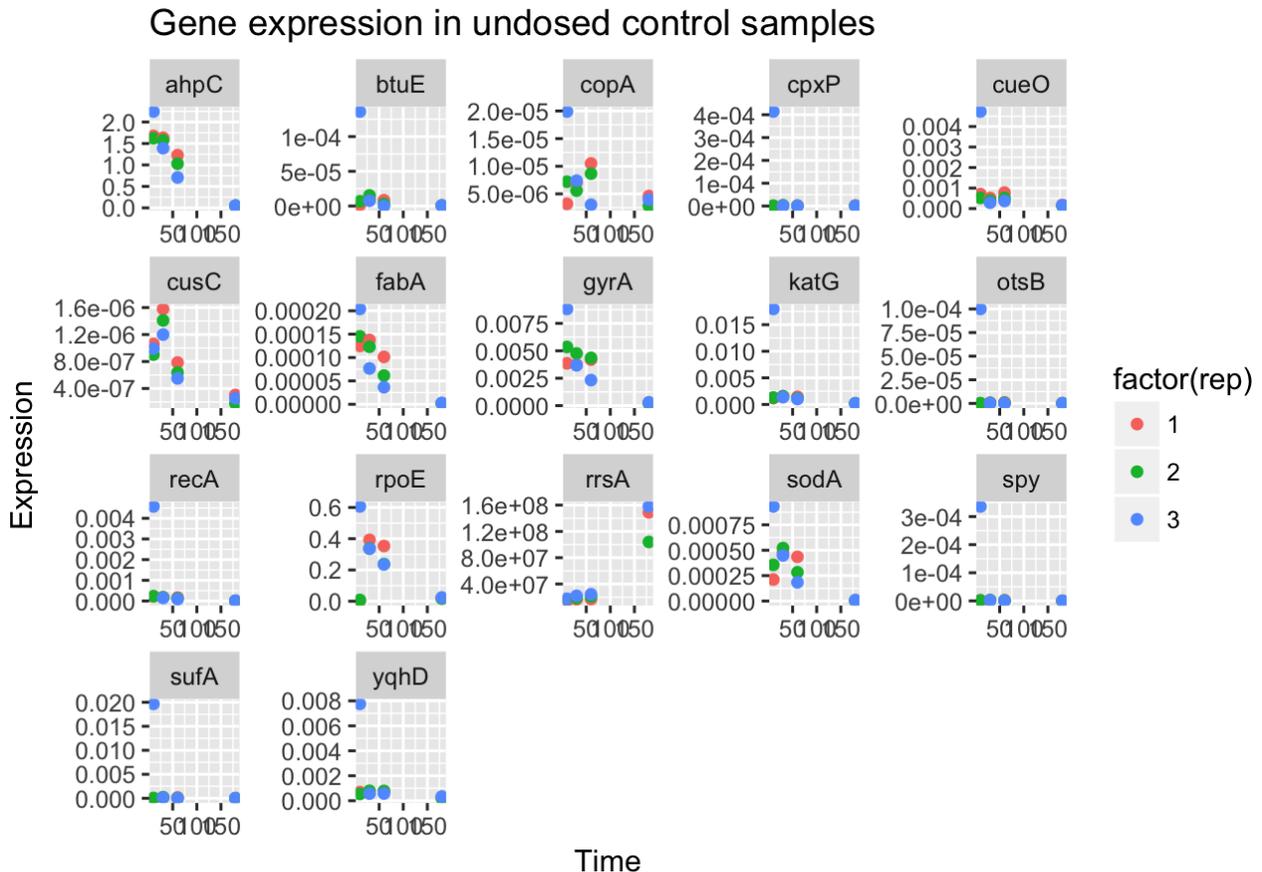
As described in the main text, the relevant dependant variable in our analysis is the “fold-control” gene expression, that is the ratio of gene expression in a treatment sample relative to the expression of that gene in the control sample.

$$\text{Fold control}_{xy} = \frac{\text{Expression}_{x,y} / \text{Expression}_{rrsA,y}}{\text{Expression}_{x,\text{Control}} / \text{Expression}_{rrsA,\text{Control}}}$$

Because control samples were not explicitly paired to treatment samples, it is necessary to average Expression_x across triplicates for normalization. However, inspection of the data showed some potential outlier data for one replicate at the $t = 10$ min. data point. An example of the undosed control data is shown below.

```
# Filter out undosed control samples to visualize gene expression
control <- dplyr::filter(gene_expression, Treatment == 'Undosed')

# These functions from ggplot2 package
control_plt <- ggplot(control, aes(x = Time,
                                  y = Expression,
                                  color = factor(rep))) +
  geom_point() +
  facet_wrap(~Gene, scales = 'free') +
  labs(title = 'Gene expression in undosed control samples')
print(control_plt)
```



Closer examination of the dataset show that the third replicate at $t = 10$ minutes of the undosed sample produced the preponderance of extreme values in this dataset as measured by absolute deviation from the median of the replicates. As such, this data point was removed from the analysis discussed in the manuscript, however pieces of the analysis with and without that data point are included here.

$$\epsilon = \frac{|Expression_x - median(Expression_x)|}{median(Expression_x)}$$

```
# Add absolute deviation from the median for outlier identification
gene_expression %<>% dplyr::group_by(Treatment, Time, Gene) %>%
  dplyr::mutate(abs_dev = abs(Expression - median(Expression)) / median(Expression)) %>%
  dplyr::ungroup()

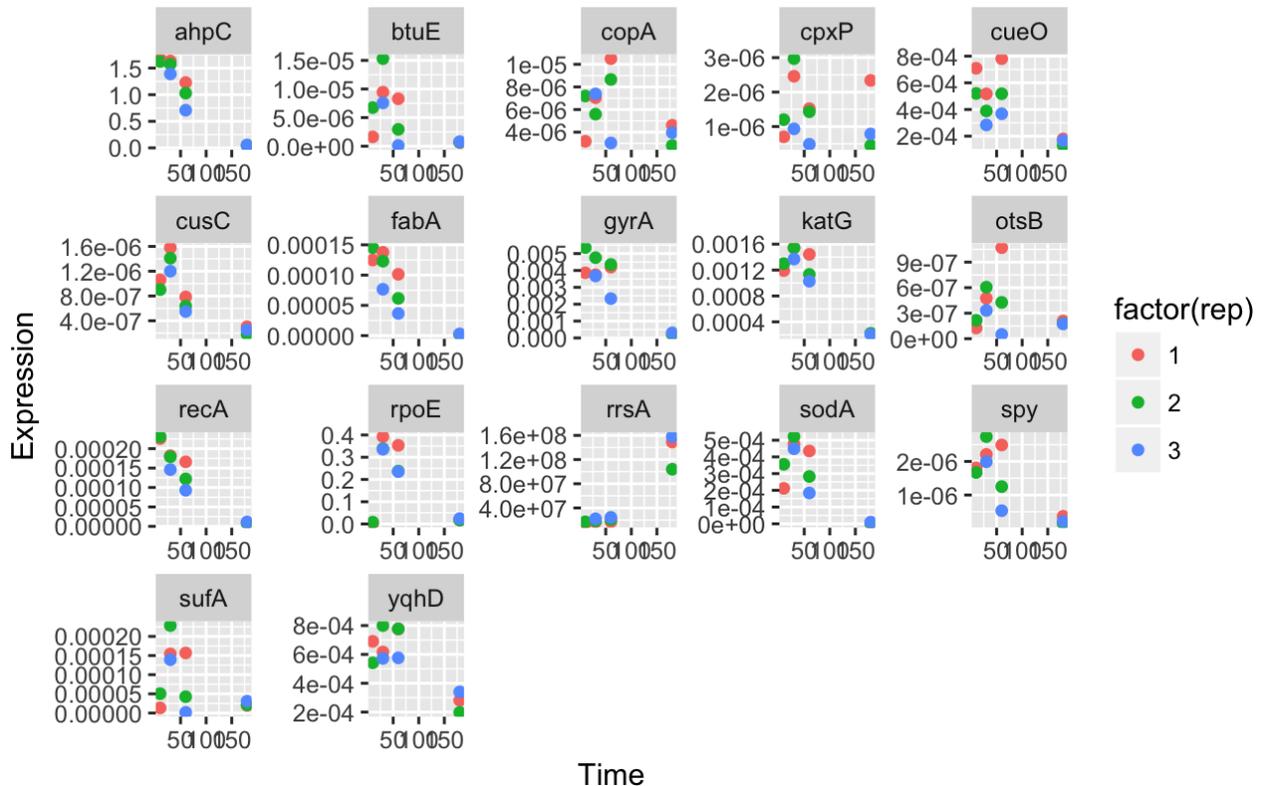
# arrange and display
gene_expression %>%
  dplyr::arrange(desc(abs_dev)) %>%
  dplyr::tbl_df()
```

```
## # A tibble: 799 x 6
##   Treatment Time Gene Expression rep abs_dev
##   <fctr> <int> <fctr> <dbl> <int> <dbl>
## 1 Undosed 10 otsB 0.000100000 3 459.82949
## 2 Undosed 10 sufA 0.019673469 3 388.57364
## 3 Undosed 10 cpxP 0.000414000 3 344.00000
## 4 Undosed 10 spy 0.000336000 3 184.63536
## 5 Undosed 10 rpoE 0.606530612 3 73.74280
## 6 ENM 10 otsB 0.000002690 1 59.99773
## 7 ENM 10 sufA 0.000075000 1 44.18072
## 8 ENM 10 btuE 0.000006410 1 23.65385
## 9 Undosed 10 btuE 0.000136000 3 19.14815
## 10 Undosed 10 recA 0.004573526 3 18.79881
## # ... with 789 more rows
```

```
# Remove control replicate with anomalous data
gene_expression_clean <- gene_expression %>%
  filter(!(Treatment == 'Undosed' & rep == 3 & Time == 10))

control_plt_clean <- ggplot(filter(gene_expression_clean, Treatment == 'Undosed'),
  aes(x = Time,
      y = Expression,
      color = factor(rep))) +
  geom_point() +
  facet_wrap(~Gene, scales = 'free') +
  labs(title = 'Gene expression in undosed control samples\nafter outlier removal')
print(control_plt_clean)
```

Gene expression in undosed control samples after outlier removal



These control data were then averaged for each gene and time point for fold-control normalization of treatment data.

```
# Average over control replicates for determining fold control
control_av <- gene_expression_clean %>%
  dplyr::filter(Treatment == 'Undosed') %>%
  dplyr::group_by(Time, Gene) %>%
  dplyr::summarize(Expression = mean(Expression)) %>%
  dplyr::ungroup()

control_av_raw <- gene_expression_clean %>% # including anomalous data
  dplyr::filter(Treatment == 'Undosed') %>%
  dplyr::group_by(Time, Gene) %>%
  dplyr::summarize(Expression = mean(Expression)) %>%
  dplyr::ungroup()

# Add fold control calculation
treatment <- dplyr::filter(gene_expression_clean, Treatment != 'Undosed') %>%
  dplyr::left_join(control_av, by = c('Time', 'Gene'),
    suffix = c('.treat', '.cont')) %>%
  dplyr::mutate(fold_control = Expression.treat/Expression.cont)

treatment_raw <- dplyr::filter(gene_expression, Treatment != 'Undosed') %>%
  dplyr::left_join(control_av, by = c('Time', 'Gene'),
    suffix = c('.treat', '.cont')) %>%
  dplyr::mutate(fold_control = Expression.treat/Expression.cont)
```

Hypothesis testing of gene expression data

The first analysis we performed on the gene expression data sought to determine if ENM exposure was sublethal by examining fold changes in *rrsA* expression at each time point. This can be done with a one-sided Mann-Whitney test:

$$H_0 : \mu_{ENM-Ctrl} = 0$$

$$H_1 : \mu_{ENM-Ctrl} < 0$$

```
# Establish sub-lethal Cu exposure using Mann-Whitney on raw
# expression of rrsA `alternative` set to "less" b.c. want one
# sided test if growth is significantly less in the treated
# samples
gene_expression_clean %>%
  dplyr::filter(Gene == 'rrsA') %>%
  dplyr::select(rep, Time, Treatment, Expression) %>%
  tidyr::spread(Treatment, Expression) %>%
  tidyr::gather(treatment, expression, ENM:Pulse) %>%
  dplyr::group_by(Time, treatment) %>%
  dplyr::do(
    broom::tidy(
      wilcox.test(x = .$expression, y = .$Undosed,
                  alternative = 'less', exact = T)
    )
  ) %>%
  dplyr::select(Time, treatment, p.value) %>%
  tidyr::spread(treatment, p.value)
```

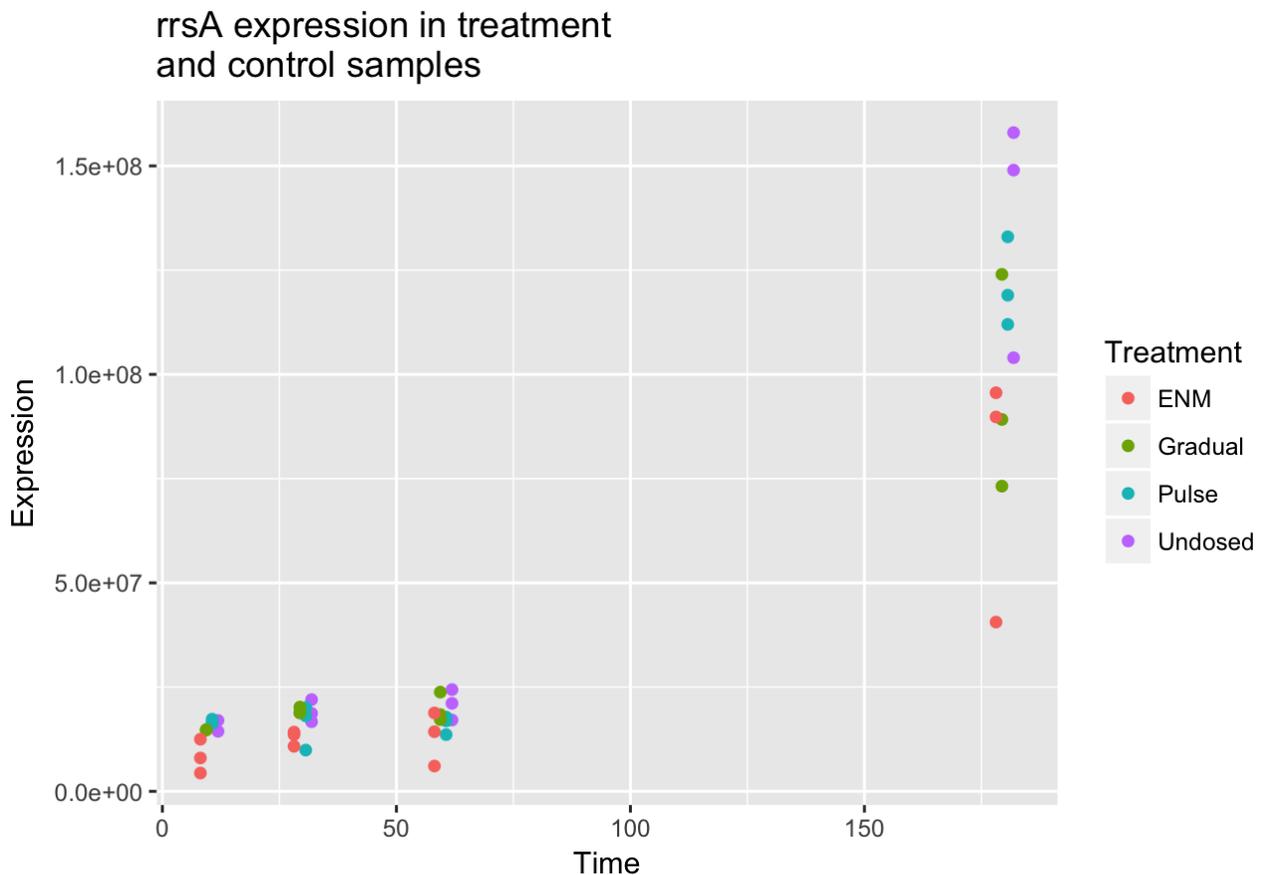
```
## # A tibble: 4 x 4
## # Groups:   Time [4]
##   Time ENM Gradual Pulse
## * <int> <dbl> <dbl> <dbl>
## 1    10  0.10 0.6666667 0.8128703
## 2    30  0.05 0.8000000 0.3500000
## 3    60  0.10 0.5000000 0.1000000
## 4   180  0.05 0.1000000 0.3500000
```

```
gene_expression %>% # including anomalous data
  dplyr::filter(Gene == 'rrsA') %>%
  dplyr::select(rep, Time, Treatment, Expression) %>%
  tidyr::spread(Treatment, Expression) %>%
  tidyr::gather(treatment, expression, ENM:Pulse) %>%
  dplyr::group_by(Time, treatment) %>%
  dplyr::do(
    broom::tidy(
      wilcox.test(x = .$expression, y = .$Undosed,
                  alternative = 'less', exact = T)
    )
  ) %>%
  dplyr::select(Time, treatment, p.value) %>%
  tidyr::spread(treatment, p.value)
```

```
## # A tibble: 4 x 4
```

```
## # Groups:   Time [4]
##   Time  ENM Gradual Pulse
## * <int> <dbl> <dbl> <dbl>
## 1   10  0.05   0.4  0.50
## 2   30  0.05   0.8  0.35
## 3   60  0.10   0.5  0.10
## 4  180  0.05   0.1  0.35
```

```
ggplot(
  filter(gene_expression_clean, Gene == 'rrsA'),
  aes(x = Time, y = Expression, color = Treatment)
) +
  geom_point(position = position_dodge(5)) +
  labs(title = 'rrsA expression in treatment\nand control samples')
```



Next we sought to determine the genes identified in the literature as being Cu-induced exhibited differential expression between undosed and ENM dosed experiments. These genes were: *copA*, *cueO*, *cusC*, *cpXP*, *sufA*, and *spy*.

```
induced_genes <- c("copA", "cueO", "cusC", "cpXP", "sufA", 'spy')

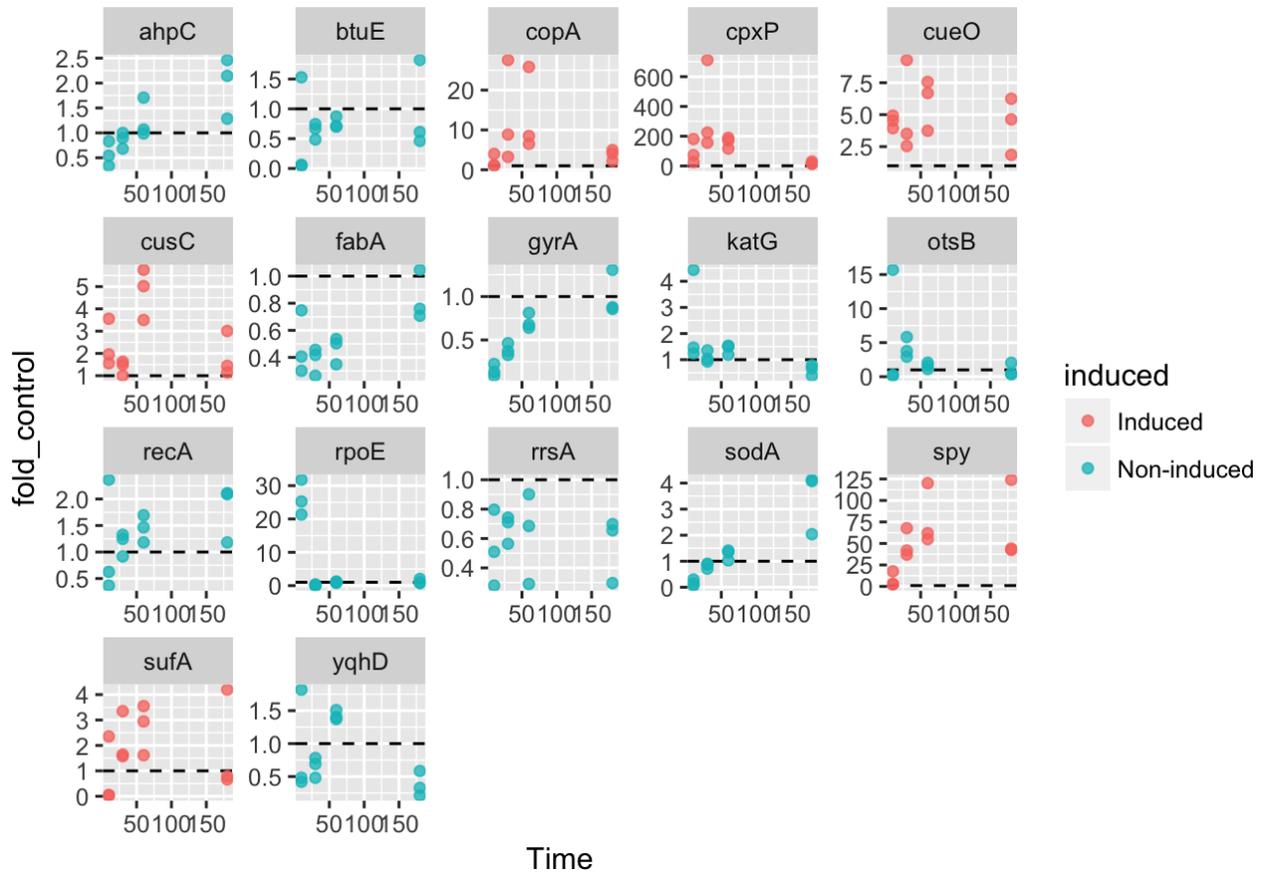
# Add identifier of Cu-induced genes
treatment %<>% dplyr::mutate(induced = ifelse(Gene %in% induced_genes,
                                           'Induced',
                                           'Non-induced'))

ggplot(filter(treatment, Treatment == 'ENM'),
```

```

aes(x = Time, y = fold_control, color = induced)) +
geom_hline(yintercept = 1, linetype = 2) +
geom_point(alpha = 0.75) +
facet_wrap(~Gene, scales = 'free')

```



```

# `alternative` set to "greater" b.c. `Treatment` factor order is
# alphabetical (i.e. ENM is group 1, Undosed is group 2)
diff_expr <- gene_expression_clean %>%
  dplyr::filter(Gene != 'rrsA', Treatment %in% c('ENM','Undosed')) %>%
  dplyr::group_by(Gene, Time) %>%
  dplyr::do(
    broom::tidy(
      wilcox.test(.$Expression ~ .$Treatment, alternative = 'greater',
        exact = T, conf.int = T)
    )
  ) %>%
  dplyr::mutate(p.value = round(p.value, 2)) %>%
  dplyr::select(Gene, Time, p.value)

diff_expr_raw <- gene_expression %>% # including anomalous data
  dplyr::filter(Gene != 'rrsA', Treatment %in% c('ENM','Undosed')) %>%
  dplyr::group_by(Gene, Time) %>%
  dplyr::do(
    broom::tidy(
      wilcox.test(.$Expression ~ .$Treatment, alternative = 'greater',
        exact = T, conf.int = T)
    )
  )

```

```

)
) %>%
dplyr::mutate(p.value = round(p.value, 2)) %>%
dplyr::select(Gene, Time, p.value)

# Combine statistical and practical data
comb_diff <- treatment %>%
  dplyr::filter(Treatment == 'ENM') %>%
  dplyr::group_by(Gene, Time) %>%
  dplyr::summarize(mu_fc = mean(fold_control)) %>%
  dplyr::ungroup() %>%
  dplyr::left_join(diff_expr, by = c('Gene', 'Time'))

comb_diff_raw <- treatment_raw %>% # including anomalous data
  dplyr::filter(Treatment == 'ENM') %>%
  dplyr::group_by(Gene, Time) %>%
  dplyr::summarize(mu_fc = mean(fold_control)) %>%
  dplyr::ungroup() %>%
  dplyr::left_join(diff_expr_raw, by = c('Gene', 'Time'))

# retain data meeting statistical and practical criterion
sig_diff <- comb_diff %>%
  dplyr::filter(p.value < 0.1, mu_fc > 2)

sig_diff_raw <- comb_diff_raw %>%
  dplyr::filter(p.value < 0.1, mu_fc > 2)

dplyr::tbl_df(sig_diff)

```

```

## # A tibble: 16 x 4
##   Gene Time mu_fc p.value
##   <fctr> <int> <dbl> <dbl>
## 1 copA 30 13.215177 0.05
## 2 copA 60 13.599099 0.05
## 3 copA 180 3.750658 0.05
## 4 cpXP 30 363.934004 0.05
## 5 cpXP 60 158.918243 0.05
## 6 cpXP 180 19.625384 0.05
## 7 cue0 30 5.109244 0.05
## 8 cue0 60 5.995792 0.05
## 9 cue0 180 4.245148 0.05
## 10 cusC 60 4.756592 0.05
## 11 otsB 30 4.168436 0.05
## 12 sodA 180 3.413932 0.05
## 13 spy 30 48.858934 0.05
## 14 spy 60 78.971683 0.05
## 15 spy 180 70.204082 0.05
## 16 sufA 30 2.188542 0.05

```

```
dplyr::tbl_df(sig_diff_raw)
```

```

## # A tibble: 17 x 4
##   Gene Time mu_fc p.value
##   <fctr> <int> <dbl> <dbl>

```

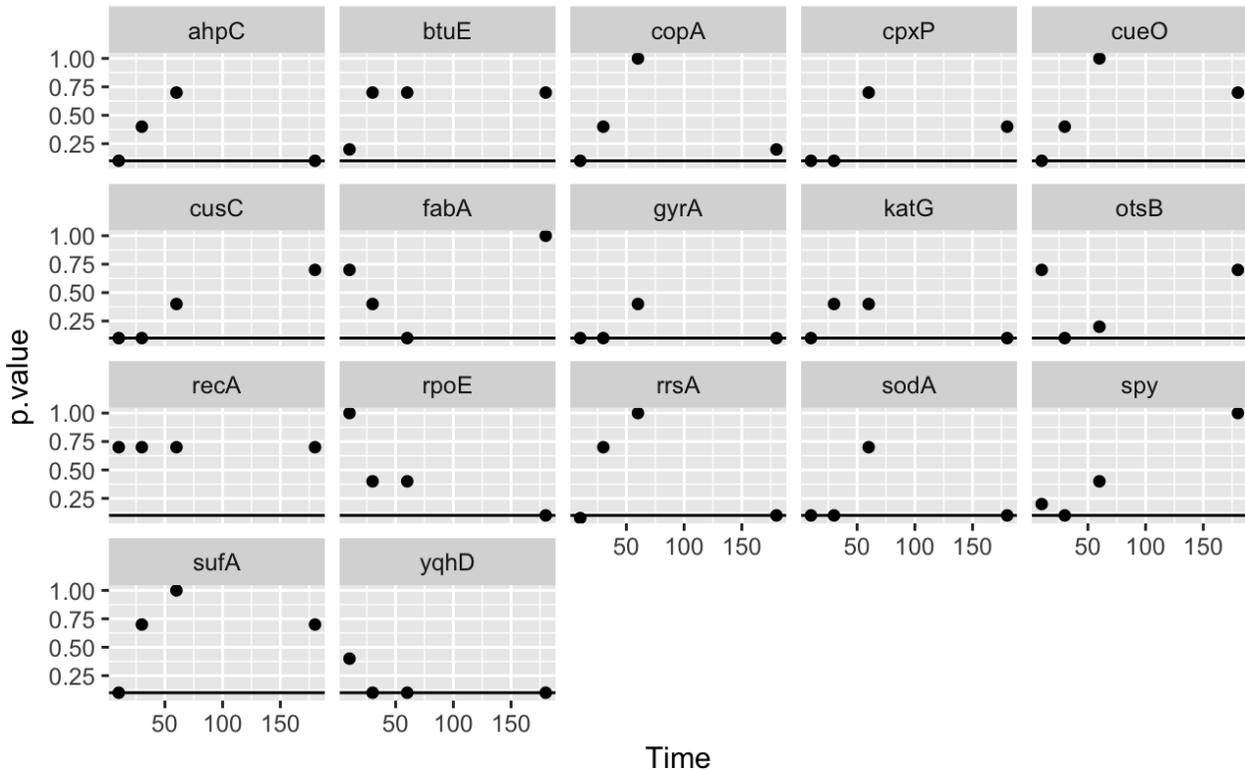
##	1	copA	30	13.215177	0.05
##	2	copA	60	13.599099	0.05
##	3	copA	180	3.750658	0.05
##	4	cpxP	30	363.934004	0.05
##	5	cpxP	60	158.918243	0.05
##	6	cpxP	180	19.625384	0.05
##	7	cueO	30	5.109244	0.05
##	8	cueO	60	5.995792	0.05
##	9	cueO	180	4.245148	0.05
##	10	cusC	10	2.358138	0.05
##	11	cusC	60	4.756592	0.05
##	12	otsB	30	4.168436	0.05
##	13	sodA	180	3.413932	0.05
##	14	spy	30	48.858934	0.05
##	15	spy	60	78.971683	0.05
##	16	spy	180	70.204082	0.05
##	17	sufA	30	2.188542	0.05

The genes which were found to be expressed significantly differently in the ENM-treated experiments compared to the undosed control ($p < 0.1$, Fold control_{x,y} > 2) were overwhelmingly in our expected list of Cu-induced genes (88% across all time points). Exclusion of the anomalous control replicate leads to no striking changes in the results of these analyses, only changing the statistical conclusion regarding *cusC* expression at $t = 10$ min.

Next, we investigated differences in gene-expression among modes of Cu exposure. This was done first through a Mann-Whitney test (two sided Wilcoxon test), between the pulse- and ENM Cu exposure modes.

```
# Mann-Whitney test for differences between ENM and Pulse Cu
# exposure modes on gene expression at each time point
treatment %>%
  filter(Treatment != 'Gradual') %>%
  group_by(Time, Gene) %>%
  do(
    tidy(
      wilcox.test(.$Expression.treat ~ .$Treatment, exact = T)
    )
  ) %>%
  ggplot(., aes(x = Time, y = p.value)) +
  geom_hline(yintercept = 0.1) +
  geom_point() +
  facet_wrap(~Gene) +
  labs(title = 'P-value of Mann-Whitney\ntest between ENM and Pulse dosing schemes')
```

P-value of Mann-Whitney test between ENM and Pulse dosing schemes



Next, a Kruskal-Wallis test, the non-parametric analog to one-way ANOVA, was used to test for differences among all three exposure modes. Results are visualized as fold control vs. time for the three Cu exposure modes, with markers dimmed for comparisons not meeting our statistical significance criteria, $\alpha \leq 0.1$.

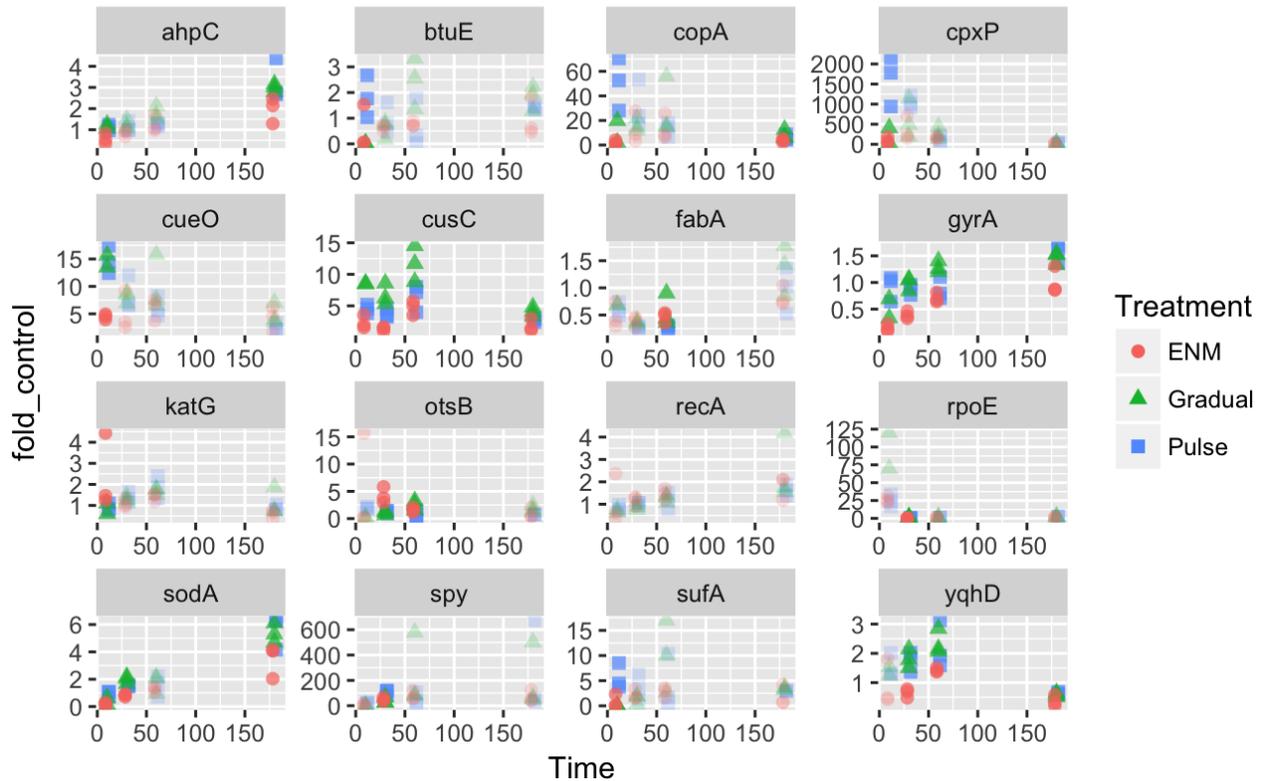
```
# Kruskal-Wallis test for differences among Cu exposure modes
# on gene expression at each time point
kw_Cu_mode <- gene_expression_clean %>%
  dplyr::filter(Treatment != 'Undosed', Gene != 'rrsA') %>%
  dplyr::group_by(Time, Gene) %>%
  dplyr::do(
    broom::tidy(
      kruskal.test(.$Expression ~ .$Treatment)
    )
  ) %>%
  dplyr::select(Gene, Time, p.value) %>%
  mutate(sig = p.value <= 0.1)

treatment %<>% dplyr::filter(Gene != 'rrsA') %>%
  dplyr::left_join(kw_Cu_mode, by = c('Gene', 'Time'))

ggplot(treatment, aes(x = Time, y = fold_control,
                      color = Treatment,
                      shape = Treatment, alpha = sig)) +
  geom_point(size = 2, position = position_dodge(5)) +
  facet_wrap(~Gene, scales = 'free') +
  scale_alpha_manual(values = c(0.2, 0.7)) +
  guides(alpha = F) +
```

```
labs(title = paste0('Gene expression vs. time for Cu exposure ',
                    'modes\nwith indication of Kruskal-Wallis ',
                    'statistical significance'))
```

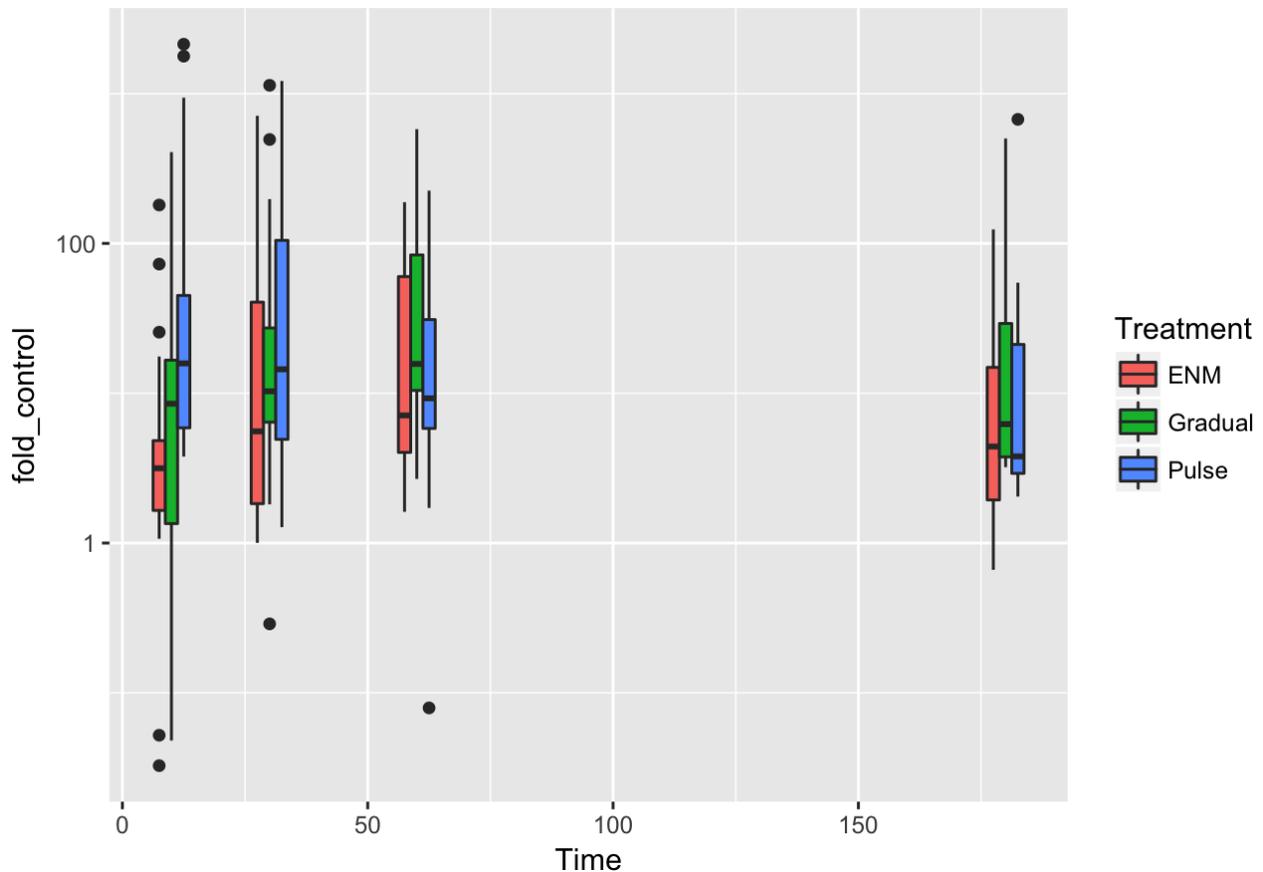
Gene expression vs. time for Cu exposure modes with indication of Kruskal-Wallis statistical significance



We further sought to explore differences among treatments on the temporal variation of gene expression in those genes that are specifically induced by Cu exposure. This is done by aggregating the fold-control data for all 6 induced genes for each treatment at each time point.

```
induced_data <- treatment %>%
  dplyr::filter(Gene %in% induced_genes)

ggplot(induced_data, aes(x = Time, y = fold_control,
                        group = interaction(Time, Treatment),
                        fill = Treatment)) +
  geom_boxplot() + scale_y_log10()
```



```
## Test across all treatments
induced_data %>%
  dplyr::group_by(Time) %>%
  dplyr::do(
    broom::tidy(
      kruskal.test(.$fold_control ~ .$Treatment)
    )
  ) %>%
  dplyr::select(Time, p.value)
```

```
## # A tibble: 4 x 2
## # Groups:   Time [4]
##   Time    p.value
##   <int>    <dbl>
## 1     10 0.008185893
## 2     30 0.419690420
## 3     60 0.137231904
## 4    180 0.269809611
```

```
## Test treatments pairwise against ENM dosing
dplyr::data_frame(vs_ENM = c('Gradual', 'Pulse')) %>%
  plyr::ddply(.(vs_ENM), function(df){
    induced_data %>%
      dplyr::filter(Treatment == 'ENM' | Treatment == df$vs_ENM) %>%
      group_by(Time) %>%
      dplyr::do(
        broom::tidy(
```

```

    wilcox.test(.$fold_control ~ .$Treatment, conf.int = T)
  )
) %>% ungroup() %>%
  dplyr::select(Time, p.value)
}) %>%
dplyr::group_by(Time) %>%
dplyr::mutate(p.value = p.adjust(p.value)) %>%
dplyr::tbl_df()

```

```

## # A tibble: 8 x 3
##   vs_ENM Time    p.value
##   <chr> <int>    <dbl>
## 1 Gradual  10 0.43949181
## 2 Gradual  30 0.46187084
## 3 Gradual  60 0.15796743
## 4 Gradual 180 0.40543981
## 5 Pulse   10 0.00254904
## 6 Pulse   30 0.34157620
## 7 Pulse   60 0.81471992
## 8 Pulse  180 0.56277743

```

There is a strong difference among treatments at the 10 minute time point (deduced from Kruskal-Wallis), attributable to the large difference between the ENM and Pulse expressions at this time point (deduced from the post-hoc, Mann-Whitney test).

Further, it was our hypothesis that, temporally, the expression of genes in response to gradual ionic Cu exposure would be more similar to the expressive response to ENM exposure than in a pulse-dosing scenario. This comparison is made using time as a continuous variable, as opposed to a series of discrete tests as before. To test this hypothesis we choose to compare the similarity (overlap) of parameter estimates of a common model form for each treatment. However, since no biologically-relevant, parametric model is available for these data we chose to fit a linear model to the fold control data with time. In order to better meet the assumptions of a linear model (specifically observation independence, uncorrelated residuals), the fold-control data are first transformed to the “lag” (or difference vs. previous time), δ_ϕ .

$$\phi = \text{Fold control}_{xy}$$

$$\delta_\phi = \phi_i - \phi_{i-1}$$

These data are repeatedly bootstrapped to generate empirical distributions of the slope and intercept of a two parameter linear model:

$$\widehat{\delta}_\phi = \beta_1 \cdot t + \beta_0$$

We can then qualitatively assess the similarities among the treatments according to this model by comparing the extent of overlap between pairs of bootstrap parameter densities.¹

```

## Transform fold-control data to lag
treat_lag <- treatment %>%
  dplyr::filter(induced == 'Induced') %>%
  dplyr::group_by(Treatment, Gene, rep) %>%
  dplyr::mutate(lag = c(NA, diff(fold_control))) %>%

```

¹Inman, H. F., & Bradley Jr, E. L. (1989). The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. *Communications in Statistics-Theory and Methods*, 18(10), 3851-3874.

```

    # NA values entered for t0 (lag not calculable), required to
    # maintain data vector dimensions.
dplyr::select(Treatment:Gene, rep, fold_control, lag) %>%
dplyr::ungroup() %>%
dplyr::filter(!is.na(lag)) # Remove NA values

## Get point estimate of model parameters
point_est <- treat_lag %>%
dplyr::group_by(Treatment) %>%
dplyr::do(
  broom::tidy(
    lm(lag ~ Time, data = .)
  )
) %>%
select(Treatment:estimate) %>% ungroup()

## Perform 1000 bootstrap estimates
## (NB: If rerunning this analysis, this may take some time)
boot_estimates <- data.frame(i = 1:1000) %>%
plyr::ddply(.,.(i), function(df){
  # Create a bootstrap instance of the data by resampling
  # completely with replacement
  resamp <- treat_lag %>%
    dplyr::group_by(Treatment, Time) %>%
    dplyr::slice(sample(1:n(), replace = T))

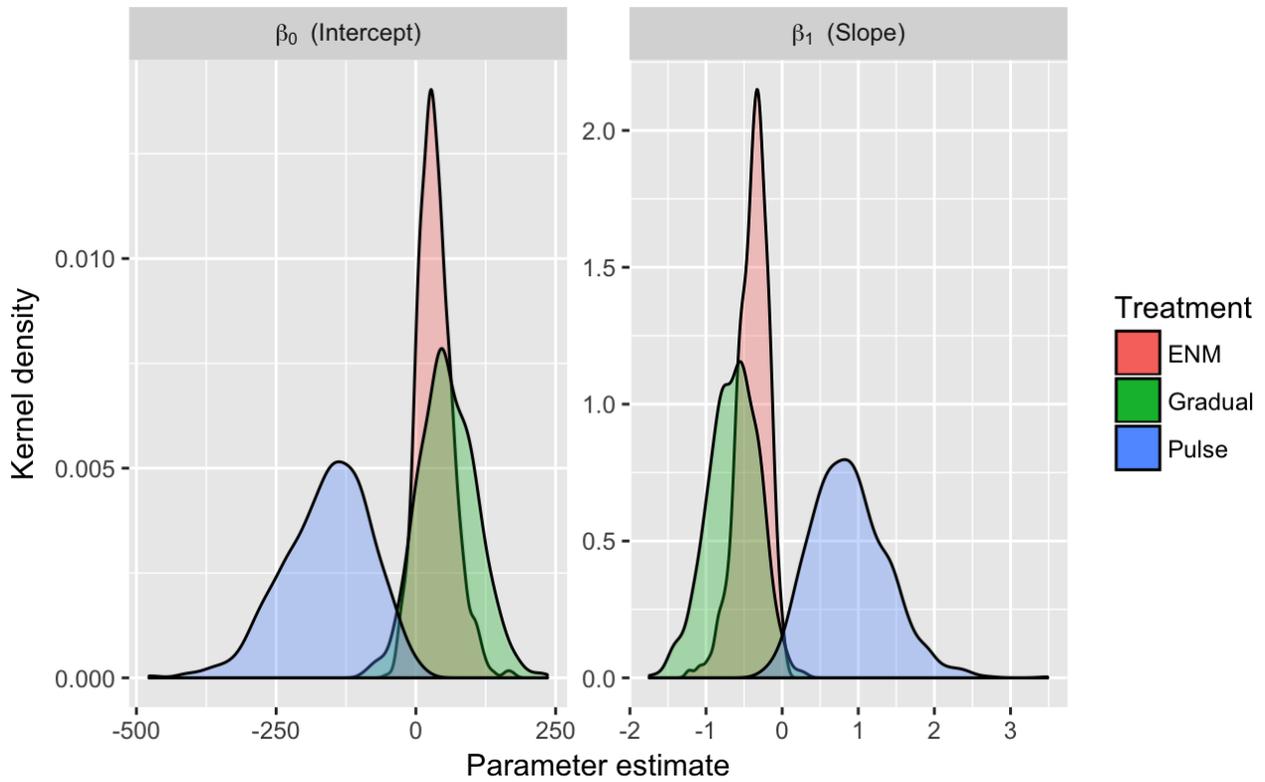
  # Fit linear model and extract parameters
  resamp_lm <- resamp %>%
    dplyr::group_by(Treatment) %>%
    dplyr::do(
      broom::tidy(
        lm(lag ~ Time, data = .)
      )
    ) %>%
    dplyr::select(Treatment:estimate)

  return(resamp_lm)
}) %>%
dplyr::mutate(term = factor(term,
  labels = c(bquote(beta[0]~" (Intercept)"),
             bquote(beta[1]~" (Slope)"))
))

param_plt <- ggplot(boot_estimates,
  aes(x = estimate, fill = Treatment)) +
  stat_density(position = 'identity', alpha = 0.33, color = 'black') +
  facet_wrap(~term, scales = 'free', labeller = label_parsed) +
  scale_x_continuous('Parameter estimate') +
  scale_y_continuous('Kernel density') +
  guides(fill = guide_legend(override.aes = list(alpha = 1)))

print(param_plt)

```



The higher degree of overlap between estimates of both parameters for ENM and Gradual (vs. Pulse), supports the stated hypothesis that the temporal response of gene expression in response to ionic copper exposure better represents that of ENM exposure than does a pulse exposure. However, our data show also that this means a mild practical difference at later time points, with early responses driving much of the overall difference.