Visualizing murine placental extracellular vesicle data with tidyNano: a computational framework for analyzing and visualizing nanoparticle data in R

Sean L Nguyen[1,2], Jacob W Greenberg[3], Hao Wang[4], Benjamin W Collaer[5], Jianrong Wang[4], Margaret G Petroff[3,6]

[1] Cell and Molecular Biology Program, Michigan State University, East Lansing, Michigan, United States of America

[2] Institute for Integrative Toxicology, Michigan State University, East Lansing, Michigan, United States of America

[3] Department of Microbiology and Molecular Genetics, Michigan State University, East Lansing, Michigan, United States of America, Michigan State University, East Lansing, Michigan, United States of America

[4] Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, Michigan, United States of America,

[5] Department of Biochemistry and Molecular Biology, Michigan State University, East Lansing, Michigan, United States of America

[6] Department of Pathobiology Diagnostic Investigation, Michigan State University, East Lansing, Michigan, United States of America

*Corresponding author

E-mail:petrof10@cvm.msu.edu (MGP)

Abstract

Extracellular vesicles (EVs) are increasingly recognized as important mediators of intercellular communication, and in mammals are generally classified as ~50-150nm exosomes, ~100-1000nm microvesicles, and apoptotic bodies, each arising as a result of different biological processes. EVs carry protein, lipids, and nucleic acids within the circulation, to target cells whereupon they mediate physiological changes. Due to their small size, quantification and characterization by conventional microscopy is not possible. However, nanoparticle tracking analysis (NTA) has provided a method to determine the fluid concentration and size of extracellular vesicles and other nanoparticles. While NTA provides statistical summaries of samples in an experiment, a recurring difficulty is the organization, manipulation, and management of raw particle count data because of the large size of datasets, and resultant vulnerability to user error. To address these limitations, we developed tidyNano, an R package that provides functions to import, clean, and quickly summarize raw NanoSight (Malvern Panalytical) data for efficient calculation of statistics and visualization. Here, we provide a framework for importing raw nanoparticle data and provide functions to facilitate rapid and efficient analysis, visualization and calculation of summary statistics. tidyNano was used to analyze murine plasma extracellular vesicles across gestation by aggregating and summarizing samples based on technical, biological and gestational parameters. In addition, we developed shinySIGHT, a Shiny web application that allows for interactive exploration and visualization of EV data. Using this package, we analyzed data generated from 36 samples of EV derived from the plasma of mice across gestation.

Peripheral EV concentration increased linearly across pregnancy, with trending increases as early as gestation day (GD) 5.5 and significant rises at GD14.5, and 17.5 relative to EV concentrations in nonpregnant females. Thus, the data highlight the utility of the mouse as a model of EV biology in pregnancy. Further, the package provides a mechanism for seamless analysis of EV data generated by NanoSight. Importantly, this package provides a straightforward framework by which diverse types of large datasets can be simply and efficiently analyzed. tidyNano and shinySIGHT are freely available under the MIT license and is hosted on GitHub (https://nguyens7.github.io/tidyNano/).

Introduction

Extracellular vesicles (EVs) encompass a broad class of membrane enclosed structures secreted by cells and are classified based on their size and subcellular origin. Exosomes, which range from 50-150nm, arise from the inward budding of late endosomes, and microvesicles, which range from 100-1000nm, develop as a result of outward budding of the plasma membrane. Because EVs have the capacity to induce physiological responses in recipient/target cells, they are of immense interest for many life science disciplines including immunology, cancer biology, and medicine [1–4]. Molecular contents of EVs in the form of lipids, proteins, and nucleic acids serve as the mechanism for intercellular communication [5]. Within the context of pregnancy, the conceptus-derived placenta secretes copious amounts of EVs that are detectable in maternal blood as early as the first trimester in women [6–8]. Placental EVs contribute to critical processes in pregnancy including vascular development of the maternal-fetal

interface and establishment of maternal immune tolerance to the antigenically foreign fetus [1,9]. Further, because placental EVs circulate in maternal peripheral blood, they can serve to provide a noninvasive 'liquid biopsy' indicating fetal health in utero; similar concepts are applied to overall health as a potential diagnostic for cancer [3,10].

While interest in EVs has burgeoned for these reasons over the last decade, their isolation and analysis has posed a number of technical challenges. Because of their small size, direct observation, characterization, and quantification of isolated EVs and other nanoparticles require specialized equipment and generates large amounts of data. One method of measuring the size and concentration of nanoparticles in solution is nanoparticle tracking analysis (NTA; NanoSight, Malvern Instruments, USA), which makes use of a microfluidic system and laser, together with a camera and software to track individual particles, measuring their size. On the basis of Brownian motion, the size and concentration of EVs are quantified [11,12]. Following sample measurement with the NanoSight, a PDF report is generated containing a line graph depicting size distribution and particle concentration, as well as summary statistics including mean, median, and mode size of the samples (S1 Fig). For further analysis, users can access the raw data, which includes particle counts, instrument settings, acquisition parameters including dilution factor, and other information.

Experiments with multiple conditions require the use of independent spreadsheet software (e.g., Microsoft Excel) for extraction of raw counts, as well as sample column header information. Spreadsheets can facilitate this process by providing user-friendly, intuitive access to automated calculations through a point-and-click interface. However,

this approach is limited in that the user interacts directly with the raw data, increasing the chances for user-introduced error [13]. In fact, Excel has been known to inadvertently change underlying raw data (e.g., changing a gene named OCT4 to the date 10-04) which can become problematic for downstream analysis [13–15]. In addition, spreadsheet software is limited to user-specific pipelines and cannot be easily expanded to accommodate multiparameter experiments. Further, the amount of work required to perform calculations typically increases linearly with additional data. Analysis of large number of samples would benefit from a computational approach but requires experiment-specific scripts which are not easily adaptable to complex experimental designs or conducive for reproducible research [16,17]. With these considerations, it becomes clear that a unified framework to process and restructure NTA count data into a consistent and predictable format is needed for downstream visualization and analysis.

We developed tidyNano, an R package that allows users to quickly import and transform data into a format suitable for statistical calculations and graphical visualizations. We also created an R Shiny web application, shinySIGHT, for further exploration of the data using a graphical user interface that can facilitate the process of inspecting and visualizing data without requiring the user to code. TidyNano provides a general framework to analyze NTA data by providing functions that address three key components of analysis: data import and cleaning, data summarization within groups, as well as visualization and calculation of statistics.

Design and Implementation

Although rearrangement and restructuring of data to a format conducive for analysis is often time-consuming and tedious, it is necessary to facilitate downstream analyses such as normalization of samples, visualization, and calculation of statistics. A recent paradigm in data cleaning known as data 'tidying', or the process of restructuring data into a format known as 'tidy' data, has become increasingly popular due to its consistent format that is amenable to data exploration and analysis [18]. Tidy data is defined as data that is organized into a format such that observations or events are represented in rows, and variables are represented in a single column. The advantage of transforming data into a 'tidy' format is that it allows for rapid subsetting, calculation of statistics, and visualization. Several popular packages exist that accept tidy data as an input allow for rearranging data, efficient data manipulation, functional programming, and visualization [19–22]. These packages share a similar philosophy, application programming interface, consistent syntax, predictable data structure, and compatible with other packages.

The core function of tidyNano is to import raw NanoSight data into a data frame, suitable for efficient analysis by rearranging it into a 'tidy' format in which observations are represented in rows and variables in a single column [18]. Data that is formatted in this manner is conducive for manipulation with the dplyr package [20], visualizations with the ggplot2 package [21], and computation with base R statistics [23] (Fig 1). tidyNano provides functions that simplify the process of importing NTA data into the R environment and offers functions to rearrange the data from the canonical 'wide' to 'long' format that is characteristic of 'tidy' data (Fig 2, 3(A), 3(B)). In addition to

143 reformatting data, tidyNano can parse out sample information and perform back
144 calculations to account for sample dilution during NTA measurement. Once in this 'tidy'
145 format, tidyNano provides functions to aggregate and calculate summary statistics that
146 are immediately suitable for further analysis, visualization with ggplot2, shinySIGHT or
147 statistical calculations (Fig 1). Functions within tidyNano make use of dplyr and tidyr
148 packages to transform and aggregate data. This allows each tidyNano output to be
149 compatible with a wide variety of R packages and suitable for recent changes in
150 visualization paradigms [19,20,23–26].

151 **Fig 1. Schema of tidyNano framework.** tidyNano (purple) is designed to facilitate the
152 process of importing and formatting data into a tidy format, such that the data are
153 compatible with existing visualization and statistical packages (cyan).

154 Data import and cleaning

155 NTA data are exported in the form of a .CSV file containing raw sample count as well as
156 instrument and parameter settings, which makes importing into the R environment with
157 standard commands (i.e., read.csv() / read_csv()) problematic. NTA data can be
158 imported into R by manually extracting tabular count data into a .CSV file, making
159 computation and analysis of data cumbersome for experiments involving many
160 samples. tidyNano makes use of the tidyr and dplyr packages to import, rearrange, and
161 format data for suitable downstream analysis [19]. These functions allow the import of
162 individual NTA files or combination of multiple files into a data frame and facilitates the
163 process of rearranging and parsing information from sample names. Additionally, these

164 functions will perform back calculations to account for dilution factors and convert
165 columns to the correct object class for downstream analysis.

166 **nanoimport and nanocombine.** tidyNano provides nanoimport() and nanocombine()
167 functions to import individual experiment summary files or combine multiple experiment
168 files into a single data frame within the R environment. Both functions are extensible to
169 accommodate user specified bin widths, particle ranges, and multiple versions of NTA
170 software. The functions automatically determine the number of samples in an NTA
171 experiment, extracts sample names and particle counts for each sample and returns a
172 single data frame.

173 **nanotidy.** nanotidy() makes use of the gather() function from the tidyr package to
174 convert the data from 'wide' to 'long' format, i.e., the condensing of individual sample
175 columns into key-value pairs [18]. The function then splits the sample column into
176 multiple columns based on user-specified names (i.e., 2018_1000_1), and performs
177 back calculations to determine the true count of particles in the original sample. Finally,
178 the column headers are converted to the correct class (e.g., categorical factor or
179 numeric). The output of nanotidy is a 'tidy' data frame that is suitable for immediate
180 visualization using existing packages such as ggplot2 or the shiny web application
181 shinySIGHT within tidyNano (Fig 2).

182 **Fig2. Example workflow of tidyNano for analysis of NTA data.** Core functions of
183 tidyNano (violet) are to facilitate extraction, formatting and aggregation of NTA data.
184 Following data import, tidyNano functions can be easily visualized using existing
185 packages such as ggplot2 (cyan) or with the interactive web application shinySIGHT.

Data summarization within groups

tidyNano provides functions for systematic calculation of summary statistics within groups which is useful for multifaceted NTA data containing multiple treatments, experimental conditions, biological replicates, and technical replicates. tidyNano summarization functions make use of tidy evaluation within the dplyr package to capture expressions for later evaluation within the R environment [20]. More specifically, these functions build upon group_by() and summarize() functions but offer greater flexibility and extensibility to accommodate multiple grouping arguments. Finally, tidyNano functions allow for dynamic naming of column prefixes, and functions can be used sequentially.

nanolyze and nanocount. The nanolyze() function calculates summary statistics by group and generates a data frame that includes the number of samples within groups, mean, standard deviation, and standard error. Further, it calculates a wide variety of statistics such as technical and biological replicate means or differences between other experimental conditions. Nanolyze() contains arguments for specifying prefixes to summary statistic columns which allows for the function to be used sequentially to average replicates such as technical, biological, and group/treatment replicates. Each iteration of nanolyze returns a tidy data frame suitable for visualization (Fig 2). The nanocount function allows for rapid calculation of the total sum of particles within groups of samples and can be combined with existing functions such as filter() to subset data.

Data visualization and statistical analysis

In addition to offering tools for data import and aggregation, tidyNano can also be used for visualization and statistical analysis. Functions within tidyNano provide a consistent output of tidy data frames which can be graphed using existing visualization packages such as ggplot2 [22].

shinySIGHT. shinySIGHT is web application built within the R shiny framework that allows the user to upload, interact with, and visualize the NanoSight data without needing to program. shinySIGHT is designed to facilitate the process of dynamically exploring NTA data and provide better understanding of particle size distribution and concentration with user adjustable sliders to specify nanometer (nm) ranges (Fig 2). shinySIGHT is available within the tidyNano package and can be run locally by using the shinySIGHT() command. Additionally, shinySIGHT is available online for free and can be accessed at https://github.com/nguyens7/shinySIGHT.

Fig 3. Data import and reformatting with nanoimport() and nanotidy(). (A) Output from nanoimport() or manually-cleaned NTA data. (B) Intermediate step of nanotidy() function which converts data from 'wide' to 'long' format, generating tidy data that can be easily filtered to isolate individual sample values. (C) Finalized output from nanotidy() function which separates 'Sample' column into user-specified columns. (D) Representative visualization of technical replicates of nanoimport() output which includes multiple sample injections from one sample with ggplot2, lines represent technical replicate measurement.

Results and Discussion

To test the package, we used tidyNano to analyze two NanoSight datasets. First set is comprised of polystyrene fluorescent and non-fluorescent bead analysis (S2 Fig) and a second set consists of a time course study of peripheral exosomes from C57BL/6 female mice across six time points throughout pregnancy. Additional information about these datasets are available in the supporting information.

Data Import and Cleaning

To demonstrate the nanoimport() function, we imported raw NTA data from a dataset of polystyrene nano beads. nanoimport() was able to extract column header and individual particle count data and combine them into a large data frame regardless of the number of samples (S3 Fig). Data can also be cleaned manually by the user in spreadsheet software and can be imported into R as was done with the murine exosome dataset (Fig 3(A)). After the NTA data frame was imported into the R environment, nanotidy() was used to reshape the data into a key-value pairs such that the individual samples and values were condensed into single columns resulting in the conversion from 'wide' to 'long' format in an intermediate step as shown in Fig 3(B). Next, nanotidy() parsed the 'Sample' column into individual (user-specified) columns and performed back calculations to obtain a 'True_count' column containing the corrected count values (Fig 3(C)). The output of nanotidy() resulted in a tidy data frame that could be summarized, visualized or manipulated for further analysis (Figs 1 and 2).

Efficient summarization

To study the concentration of peripheral exosomes in murine pregnancy, we chose to sample across four time points corresponding to different embryonic developmental stages (gestation day (GD) 4.5, 10.5, 14.5, and 17.5). The time course study was comprised of biological replicates, throughout 6 time points of gestation (n = 5-7), three technical replicates, and two injection replicates totaling 222 unique measurements (S3 Fig). To summarize technical replicate data, nanolyze() was used to systematically average three technical replicates within each biological and injection replicate and was able to return a data frame that was suitable for plotting (S3 Fig). To demonstrate the ability for nanolyze to be used sequentially, we calculated averages of two injections using technical replicate data (S4 Fig, Fig 4(A)). Nanolyze() was then repeated to determine the mean exosome concentration between biological replicates using Fig 4(A) data as input (Fig 4(B)). Nanolyze() worked sequentially due to the name argument that allows for specifying the prefix to summary column headers and param_var argument which specifies the column to perform the calculation (S1 Fig). To determine the total number of particles in a sample, we used nanocount() to calculate the concentration of particles within each biological replicate using Fig 4(A) data. Nanocount() was able to return a tidy data frame that was suitable for visualization (Fig 6(A)).

Fig 4. Multiparameter summary statistics and visualization. (A) Output from nanolyze() which calculates mean, standard deviation, and standard error within specified groups. Corresponding line graph of exosome concentration as a function of

size, split by gestational age, different colored lines within each group represent a single biological replicate. (B) Summarization of biological replicate data from nanolyze() of 3(A) data. Corresponding line graph depicting mean exosome concentration of biological replicates as a function of size, grey area surrounding lines represent standard error of the mean.

Visualization

The output from each tidyNano function is a tidy data frame conducive for graphical representation. Raw data can be visualized to determine technical replicate, and sample variation and particle size ranges can be easily examined. After import of polystyrene NTA data with nanoimport() and data reformatting with nanotidy(), the data frame output was saved as an .Rds data file using the nanosave() function. The .Rds file was then visualized using the shinySIGHT web application which dynamically created line graphs depicting fluorescent and non-fluorescent bead data (Fig 5). Documentation and a detailed vignette outlining how to use shinySIGHT are available in S1 File.

Fig 5. Interactive data manipulation and visualization with shinySIGHT web application. shinySIGHT allows users to upload tidyNano data to visualize and manipulate data using a graphical user interface. shinySIGHT automatically generates plots from user uploaded data as well as displays the underlying data frames that make up the visualizations.

With the murine pregnancy data, we were able to visualize technical replicate and injection level NTA data for all samples (S5 Fig) as well as individual samples (Fig

290 3(D)). Summarization steps from each nanolyze() application was examined by line
291 graph with ggplot2 and demonstrated mean exosome concentration of each animal split
292 by gestational time point (Fig 4(A)) as well as mean concentration of biological
293 replicates (Fig 4(B)). Following mean concentration of individual replicates with
294 nanocount(), total exosome concentration was plotted to determine the distribution of
295 exosomes across gestation (Fig 6(B)). shinySIGHT was able to visualize each output
296 from nanolyze() (S6 Fig).

297 **Fig 6. Calculation of extracellular vesicle counts and statistics with nanocount(),**
298 **nanoShapiro() and nanoTukey().** (A) nanocount() function determines the total
299 concentration of particles. (B) Boxplot of murine peripheral exosomes across
300 pregnancy, each point represents an individual biological replicate. (C) nanoShapiro()
301 function determines Gaussian distribution of data. (D) ANOVA output generated from
302 nanocount of data frame in (A). (E) nanoTukey() output of pair-wise post hoc analysis
303 comparing extracellular vesicle concentration across gestation.

304 ## Statistics

305 To determine which statistical test to use, we utilized nanoShapiro() to determine the
306 normality of the murine exosome dataset, which outputs a data frame suggesting with
307 suggestions to a perform a parametric statistical test (Fig 6(C)). Next, we used existing
308 base R statistical packages to run an ANOVA on our samples to compare total
309 exosome count to gestational day using Fig 6(A) data as our input [23]. The result of the
310 ANOVA suggested that there was a significant difference between total exosome count
311 and gestational age ($p < 0.001$) (Fig 6(D)). We then utilized the nanoTukey function to

run a Tukey post-hoc test which returned a tidy data frame of pair-wise comparisons or total exosome count between gestational ages (Fig 6(E)). There were significant differences between non-pregnant and GD14.5, non-pregnant and GD17.5, and GD4.5 and GD14.5. These results reveal that peripheral exosomes increase across gestation, peaking at GD14.5 and corresponding with placental mass.

Conclusion

We created the tidyNano package as a framework to import, tidy, and summarize nanoparticle tracking analysis data. tidyNano allows for the import of raw NTA data from multiparameter experiments with the nanoimport() function and reformatted into a tidy format with nanotidy(). Technical, biological, and group/treatment replicates can be quickly summarized with nanolyze() and particle counts can be calculated with nanocount(). tidyNano functions return tidy data frames which allow for visualization of data at each step. Further, tidyNano data frames are suitable for statistical analysis using existing R packages. Data normality can be calculated dynamically within user specified groups with nanoShapiro() and data can be nested within groups for applying functions across experimental groups/conditions. To facilitate data exploration and visualization, users can use shinySIGHT to interactively plot and manipulate data within a web application. We demonstrate how tidyNano can be used to analyze a multiparameter murine exosome dataset and determined that peripheral exosomes significantly increase from virgin and GD 14.5 and 17.5 and returned to pre-pregnancy concentrations following parturition. Future studies may seek to quantify the placental

333 EV contribution to total circulating EV population through the use of fluorescent labeling
334 of placental specific vesicles.

335 Availability and Future Directions

336 tidyNano is an open source package developed in R, hosted on GitHub
337 (https://github.com/nguyens7/tidyNano), and freely available under the MIT license.
338 Data, supporting documentation, and vignettes can be accessed at the package website
339 (https://nguyens7.github.io/tidyNano/) or GitHub
340 (https://github.com/nguyens7/tidyNano/tree/master/vignettes). tidyNano summarization
341 functions are general purpose and can be adapted to analyze other tidy data including
342 non-nanoparticle data. Additionally, tidyNano will be updated to accommodate different
343 NanoSight models and NTA software.

344 References

345

346 1. Kshirsagar SK, Alam SM, Jasti S, Hodes H, Nauser T, Gilliam M, et al.
347 Immunomodulatory molecules are released from the first trimester and term placenta
348 via exosomes. Placenta. Elsevier Ltd; 2012;33: 982–990.
349 doi:10.1016/j.placenta.2012.10.005

350 2. Jasti S, Farahbakhsh M, Nguyen S, Petroff BK, Petroff MG. Immune response to a
351 model shared placenta/tumor-associated antigen reduces cancer risk in parous mice.
352 Biology of Reproduction. 2016;96: 134–144. doi:10.1095/biolreprod.116.144907

353 3. Melo Sa, Luecke LB, Kahlert C, Fernandez AF, Gammon ST, Kaye J, et al. Glypican-
354 1 identifies cancer exosomes and detects early pancreatic cancer. Nature. 2015;523:
355 177–182. doi:10.1038/nature14581

356 4. Pi F, Binzel DW, Lee TJ, Li Z, Sun M, Rychahou P, et al. Nanoparticle orientation to
357 control RNA loading and ligand display on extracellular vesicles for cancer regression.
358 Nature Nanotechnology. Springer US; 2018;13: 82–89. doi:10.1038/s41565-017-0012-z

359 5. Van Niel G, D'Angelo G, Raposo G. Shedding light on the cell biology of extracellular
360 vesicles. Nature Reviews Molecular Cell Biology. Nature Publishing Group; 2018;19:
361 213–228. doi:10.1038/nrm.2017.125

362 6. Mitchell MD, Peiris HN, Kobayashi M, Koh YQ, Duncombe G, Illanes SE, et al.
363 Placental exosomes in normal and complicated pregnancy. American Journal of
364 Obstetrics and Gynecology. Elsevier Inc. 2015;213: S173–S181.
365 doi:10.1016/j.ajog.2015.07.001

366 7. Southcombe J, Tannetta D, Redman C, Sargent I. The immunomodulatory role of
367 syncytiotrophoblast microvesicles. PLoS ONE. 2011;6.
368 doi:10.1371/journal.pone.0020245

369 8. Sarker S, Scholz-Romero K, Perez A, Illanes SE, Mitchell MD, Rice GE, et al.
370 Placenta-derived exosomes continuously increase in maternal circulation over the first
371 trimester of pregnancy. Journal of translational medicine. 2014;12: 204.
372 doi:10.1186/1479-5876-12-204

373 9. Prabhudas M, Bonney E, Caron K, Dey S, Erlebacher A, Fazleabas A, et al.
374 Commentary Immune mechanisms at the maternal-fetal interface : perspectives and
375 challenges. Nat Immunol. 2015;16: 328–334. doi:10.1038/ni.3131

376 10. Hoshino A, Costa-Silva B, Shen T-L, Rodrigues G, Hashimoto A, Tesic Mark M, et
377 al. Tumour exosome integrins determine organotropic metastasis. Nature. Nature
378 Publishing Group; 2015;527: 329–335. doi:10.1038/nature15756

379 11. Filipe V, Hawe A, Jiskoot W. Critical evaluation of nanoparticle tracking analysis
380 (NTA) by NanoSight for the measurement of nanoparticles and protein aggregates.
381 Pharmaceutical Research. 2010;27: 796–810. doi:10.1007/s11095-010-0073-2

382 12. Dragovic RA, Gardiner C, Brooks AS, Tannetta DS, Ferguson DJ, Hole P, et al.
383 Sizing and phenotyping of cellular vesicles using Nanoparticle Tracking Analysis.
384 Nanomedicine: Nanotechnology, Biology, and Medicine. Elsevier Inc. 2011;7: 780–788.
385 doi:10.1016/j.nano.2011.04.003

386 13. Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, et al. Best
387 Practices for Scientific Computing. PLoS Biology. 2014;12.
388 doi:10.1371/journal.pbio.1001745

389 14. Zeeberg BR, Riss J, Kane DW, Bussey KJ, Uchio E, Linehan WM, et al. Mistaken
390 identifiers: Gene name errors can be introduced inadvertently when using Excel in
391 bioinformatics [1]. BMC Bioinformatics. 2004;5: 1–6. doi:10.1186/1471-2105-5-80

15. Ziemann M, Eren Y, El-Osta A. Gene name errors are widespread in the scientific literature. Genome Biology. Genome Biology; 2016;17: 17–19. doi:10.1186/s13059-016-1044-7

16. Wilson G, Bryan J, Cranston K, Kitzes J, Nederbragt L, Teal TK. Good Enough Practices in Scientific Computing. PLoS Computational Biology. 2016; 1–20. doi:10.1371/journal.pcbi.1005510

17. Sandve GK, Nekrutenko A, Taylor J, Hovig E. Ten Simple Rules for Reproducible Computational Research. PLoS Computational Biology. 2013;9: 1–4. doi:10.1371/journal.pcbi.1003285

18. Wickham H. Tidy Data. Journal of Statistical Software. 2014;59. doi:10.18637/jss.v059.i10

19. Wickham H, Henry L. tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions [Internet]. 2018. Available: https://cran.r-project.org/package=tidyr

20. Wickham H, François R, Henry L, Müller K. dplyr: A Grammar of Data Manipulation [Internet]. 2018. Available: https://cran.r-project.org/package=dplyr

21. Henry L, Wickham H. purrr: Functional Programming Tools [Internet]. 2017. Available: https://cran.r-project.org/package=purrr

22. Wickham H. ggplot2: Elegant Graphics for Data Analysis [Internet]. Springer-Verlag New York; 2009. Available: http://ggplot2.org

23. R Core Team. R: A Language and Environment for Statistical Computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2018. Available: https://www.r-project.org/

24. Robsinon D. broom: Convert Statistical Analysis Objects into Tidy Data Frames [Internet]. 2018. Available: https://cran.r-project.org/package=broom

25. Weissgerber TL, Milic NM, Winham SJ, Garovic VD. Beyond Bar and Line Graphs: Time for a New Data Presentation Paradigm. PLoS Biology. 2015;13: 1–10. doi:10.1371/journal.pbio.1002128

26. Weissgerber TL, Garovic VD, Savic M, Winham SJ, Milic NM. From Static to Interactive: Transforming Data Visualization to Improve Transparency. PLoS Biology. 2016;14: 1–8. doi:10.1371/journal.pbio.1002484

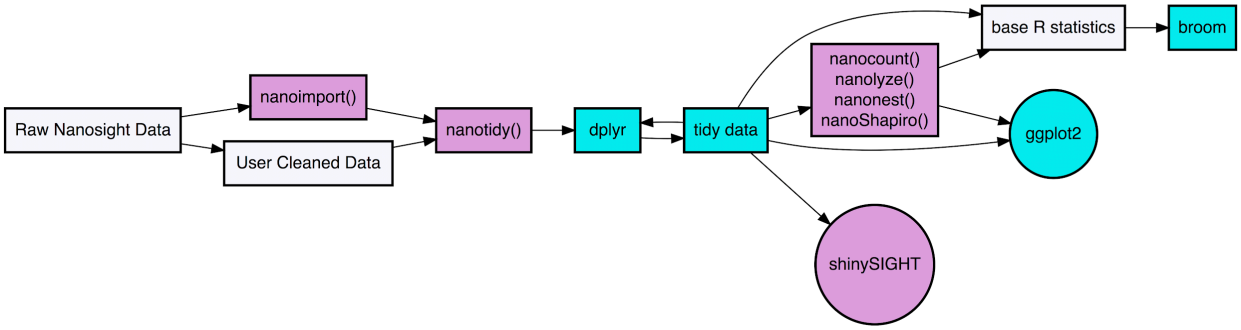


Fig1

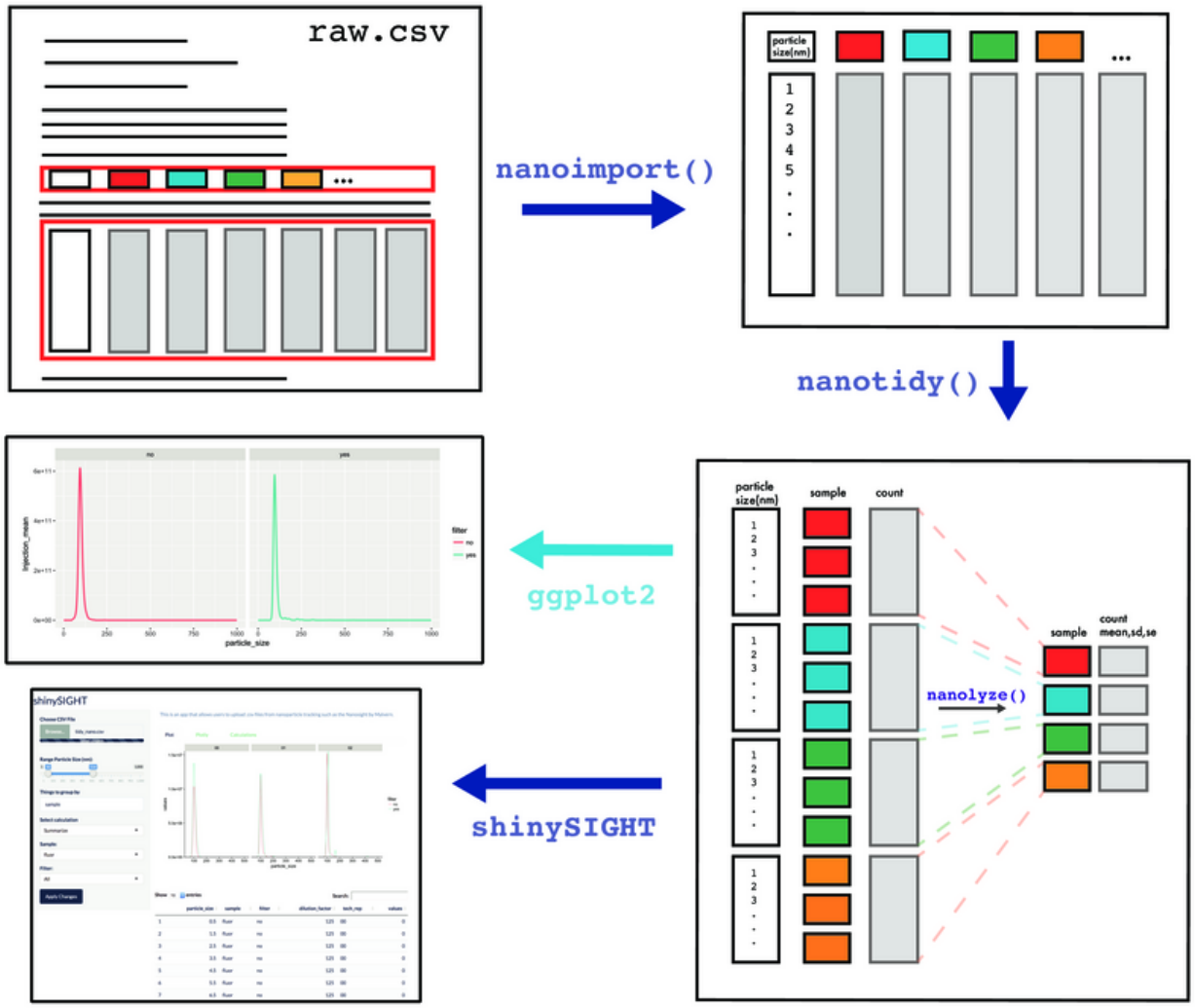

raw.csv
nanoimport()
particle size(nm)
nanotidy()
particle size(nm)
sample
count
nanolyze()
sample
count mean,sd,se
ggplot2
shinySIGHT

Fig2

A

	particle_size	2_500_1_00	2_500_1_01	2_500_1_02	2_500_2_00	2_500_2_01
1	0.5	0	0	0	0	0
2	1.5	0	0	0	0	0
3	2.5	0	0	0	0	0
4	3.5	0	0	0	0	0
5	4.5	0	0	0	0	0
6	5.5	0	0	0	0	0
7	6.5	0	0	0	0	0
8	7.5	0	0	0	0	0
9	8.5	0	0	0	0	0
10	9.5	0	0	0	0	0

1-10 of 1,000 rows | 223 columns

B

	particle_size	Sample	Count
1	0.5	2_500_1_00	0
2	1.5	2_500_1_00	0
3	2.5	2_500_1_00	0
4	3.5	2_500_1_00	0
5	4.5	2_500_1_00	0
6	5.5	2_500_1_00	0
7	6.5	2_500_1_00	0
8	7.5	2_500_1_00	0
9	8.5	2_500_1_00	0
10	9.5	2_500_1_00	0

1-10 of 222,000 rows

C

	particle_size	Sample_ID	Dilution	Injection	Tech_rep	Count	True_count
1	0.5	25	500	1	0	0	0
2	0.5	25	500	1	1	0	0
3	0.5	25	500	1	2	0	0
4	0.5	25	500	2	0	0	0
5	0.5	25	500	2	1	0	0
6	0.5	25	500	2	2	0	0
7	1.5	25	500	1	0	0	0
8	1.5	25	500	1	1	0	0
9	1.5	25	500	1	2	0	0
10	1.5	25	500	2	0	0	0

1-10 of 3,000 rows

D

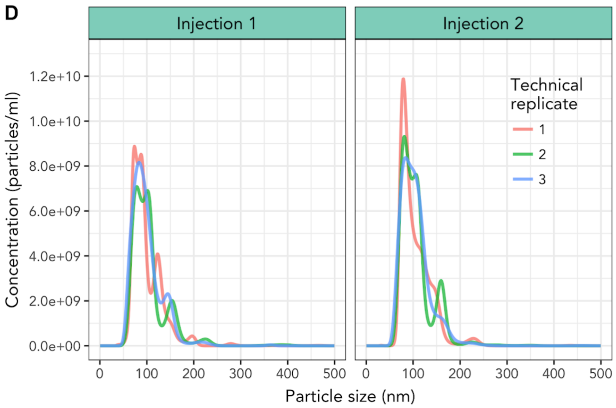
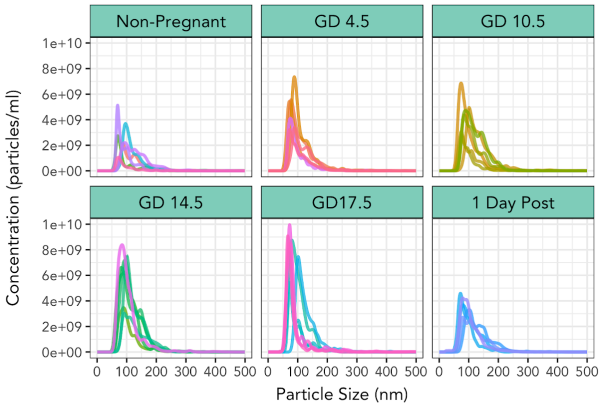


Fig 3

A

	particle_size	Sample_ID	Injection_N	Injection_mean	Injection_sd	Injection_se	Day
1	0.5	1	2	0	0	0	1
2	0.5	2	2	0	0	0	1
3	0.5	3	2	0	0	0	1
4	0.5	4	2	0	0	0	1
5	0.5	5	2	0	0	0	1
6	0.5	7	2	0	0	0	5
7	0.5	8	2	0	0	0	5
8	0.5	9	2	0	0	0	5
9	0.5	10	2	0	0	0	5
10	0.5	11	2	0	0	0	5

1-10 of 38,000 rows



B

	particle_size	Day	Day_N	Day_mean	Day_sd	Day_se
1	0.5	1	6	0	0	0
2	0.5	5	6	0	0	0
3	0.5	10	6	0	0	0
4	0.5	14	7	0	0	0
5	0.5	17	7	0	0	0
6	0.5	20	5	0	0	0
7	1.5	1	6	0	0	0
8	1.5	5	6	0	0	0
9	1.5	10	6	0	0	0
10	1.5	14	7	0	0	0

1-10 of 6,000 rows

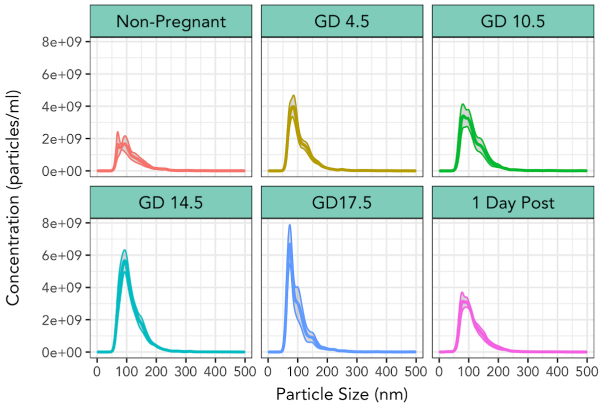


Fig 4

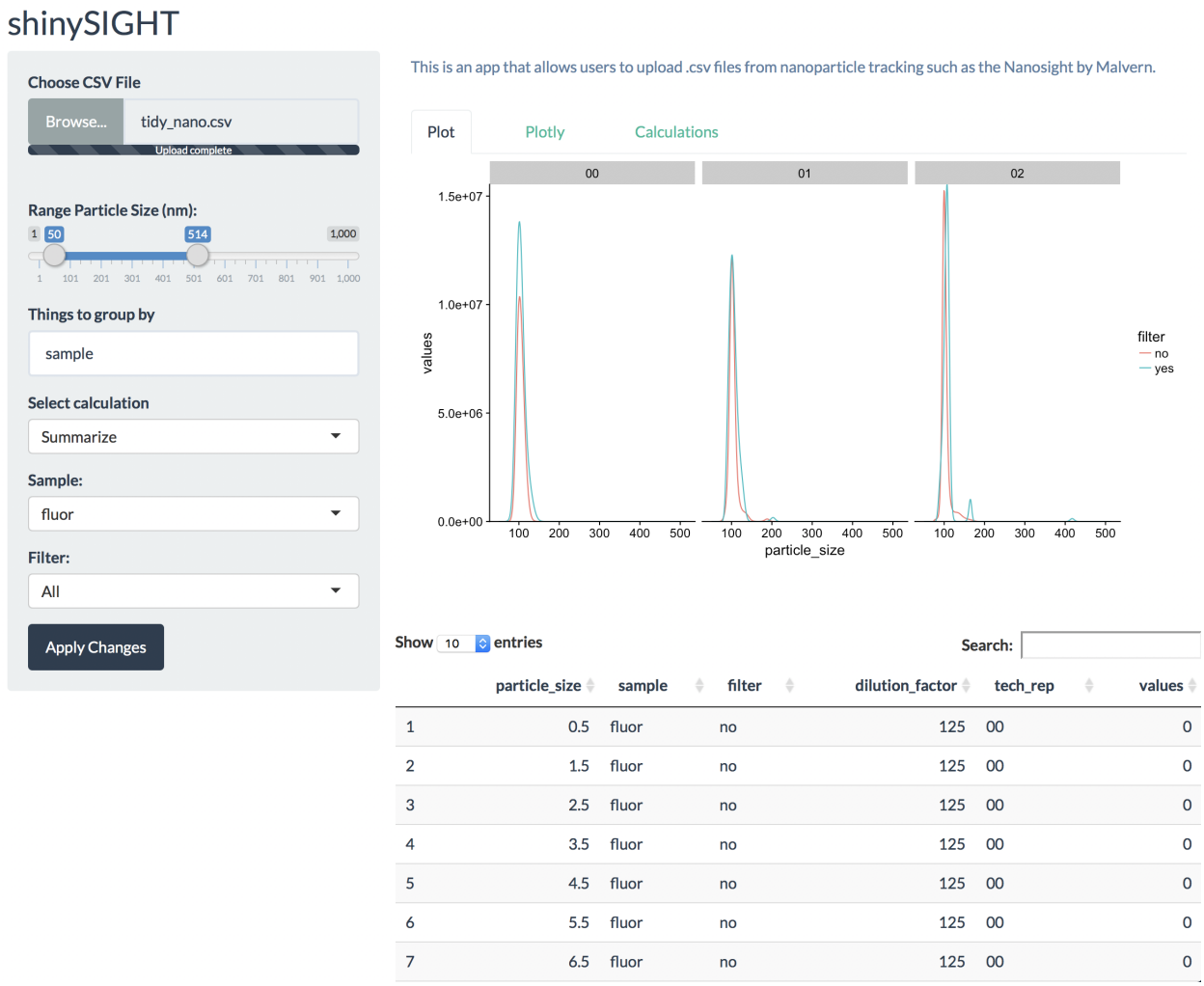

shinySIGHT
Choose CSV File
Browse...
tidy_nano.csv
Upload complete
Range Particle Size (nm):
1
50
514
1,000
1 101 201 301 401 501 601 701 801 901 1,000
Things to group by
sample
Select calculation
Summarize
Sample:
fluor
Filter:
All
Apply Changes
This is an app that allows users to upload .csv files from nanoparticle tracking such as the Nanosight by Malvern.
Plot
Plotly
Calculations
00
01
02
1.5e+07
1.0e+07
5.0e+06
0.0e+00
values
100 200 300 400 500
particle_size
filter
no
yes
Show 10 entries
Search:
particle_size sample filter dilution_factor tech_rep values
1 0.5 fluor no 125 00 0
2 1.5 fluor no 125 00 0
3 2.5 fluor no 125 00 0
4 3.5 fluor no 125 00 0
5 4.5 fluor no 125 00 0
6 5.5 fluor no 125 00 0
7 6.5 fluor no 125 00 0

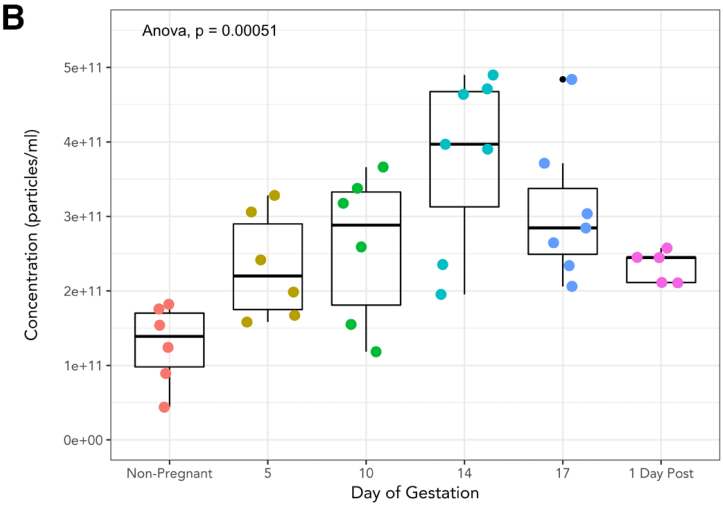
Fig 5

A

	Sample_ID	Day	Sample_N	Sample_count
1	1	1	1000	124235076333
2	10	5	1000	306075805167
3	11	5	1000	241497990000
4	12	5	1000	328132781250
5	13	10	1000	337823313000
6	14	10	1000	317599048333
7	15	10	1000	154870656917
8	16	10	1000	118335894500
9	17	10	1000	259032848167
10	18	10	1000	366183051833

1 -10 of 37 rows

B



C

	statistic	p.value	method	Normal_dist	Statistical_test
1	0.9672272	0.3383502	Shapiro-Wilk normality test	TRUE	Perform parametric test

D

	term	df	sumsq	meansq	statistic	p.value
1	Day	5	2.250291e+23	4.500582e+22	6.059435	0.000505435
2	Residuals	31	2.302493e+23	7.427396e+21	NA	NA

E

	term	comparison	adj.p.value
1	Day	14-1	0.0001568756
2	Day	17-1	0.0088825323
3	Day	14-5	0.0531415780
4	Day	20-14	0.0760557726
5	Day	10-1	0.1170207632
6	Day	14-10	0.1634542904
7	Day	5-1	0.3006484222
8	Day	20-1	0.3448108389
9	Day	17-14	0.6454840035
10	Day	17-5	0.6457892167

Fig 6