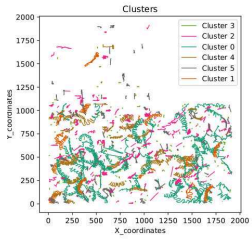
Jan 27, 2026 Version 2

Optimizing CASA Data: Transforming Sperm Coordinates into Long Format for Enhanced Machine Learning Analysis V.2



DOI
https://dx.doi.org/10.17504/protocols.io.14egn6p4ql5d/v2

Cindy Rivas Arzaluz[1], Claudia Treviño Santacruz[1], María Elena Ayala Escobar[2], Andrés Aragón Martínez[3]

[1]Instituto de Biotecnología, UNAM; [2]FES Zaragoza, UNAM; [3]FES Iztacala, UNAM

Andrés Aragón Martínez: Correspondence armandres@gmail.com;

Laboratory of Gametes ...

Cindy Rivas

OPEN ACCESS

DOI: https://dx.doi.org/10.17504/protocols.io.14egn6p4ql5d/v2

Protocol Citation: Cindy Rivas Arzaluz, Claudia Treviño Santacruz, María Elena Ayala Escobar, Andrés Aragón Martínez 2026. Optimizing CASA Data: Transforming Sperm Coordinates into Long Format for Enhanced Machine Learning Analysis. **protocols.io** https://dx.doi.org/10.17504/protocols.io.14egn6p4ql5d/v2Version created by **Cindy Rivas**

Manuscript citation:
Rodríguez-Martínez EA, Rivas CU, Ayala ME, Blanco-Rodríguez R, Juarez N, Hernandez-Vargas EA and Aragón A (2023) A new computational approach, based on images trajectories, to identify the subjacent heterogeneity of sperm to the effects of ketanserin. *Cytometry. Part A* **103** 655–663.

Protocol status: Working
We use this protocol and it's working

Created: January 27, 2026

Last Modified: January 27, 2026

Protocol Integer ID: 241655

Keywords: CASA system, clusters, sperm trajectories, algorithm, machine learning, trajectories of individual sperm, sperm coordinates into long format, assisted sperm analysis, transforming sperm coordinate, representations of sperm, individual sperm, reconstruction of motility parameter, sperm, motility parameter data, coordinates of hamster sperm, trajectory image, detected sperm, associated motility parameter, associated set of motility parameter, motility parameter, kinematic subpopulations in dataset, hamster sperm, capture routine, capture speed, trajectory, kinematic behavior, identifying kinematic subpopulation, video sequence, casa data, input for machine learning, optimizing casa data, retrieval of detailed data, detailed data, coordinate data, curvilinear velocity

Funders Acknowledgements:
UNAM-PAPIIT
Grant ID: IN224925
UNAM-PAPIIT
Grant ID: IN215425

Disclaimer

Abstract

Some Computer-Assisted Sperm Analysis (CASA) systems allow the retrieval of detailed data for each sperm analyzed during a capture routine. By "capture routine," we refer to the process of recording a video sequence for a defined period, typically one or two seconds. The data obtained include traditional kinematic or motility parameters, such as VCL (Curvilinear Velocity), VAP (Average Path Velocity), VSL (Straight-Line Velocity), LIN (Linearity), STR (Straightness), BCF (Beat Cross Frequency), and ALH (Amplitude of Lateral Head Displacement). Additional parameters may also be available, depending on the CASA system's manufacturer and software version.

Since motility parameters are derived from coordinate data, they serve as condensed representations of sperm kinematic behavior. Consequently, the coordinate data contains a wealth of additional information, enabling not only the reconstruction of motility parameters but also the trajectories followed by individual sperm. It is important to note that these trajectories cannot be reconstructed solely from motility parameters. Thus, we emphasize that each trajectory has an associated set of motility parameters (Rodríguez-Martínez et al., 2023). Despite the inherent richness of coordinate data, current methodologies for identifying kinematic subpopulations in datasets have relied exclusively on motility parameters (Ramón and Martínez-Pastor, 2018).

Coordinates can also be used to reconstruct the trajectories of individual sperm analyzed in a CASA system. These trajectory images can subsequently serve as input for machine learning algorithms, which can cluster the images into groups (subpopulations). These subpopulations can then be statistically characterized based on their associated motility parameters.

In this protocol, we describe how we constructed a coordinate dataset to serve as input for machine learning algorithms, specifically the one implemented by Rodríguez-Martínez et al. (2023). The data corresponds to coordinates of hamster sperm analyzed using a CASA system (SMAS, Version 3.18), with a capture speed set at 50 fps for one second (Fujinoki M, personal communication). Each capture routine in the SMAS system generates two files: the first contains motility parameter data, while the second contains the coordinates of the detected sperm.

The procedure comprises three stages: (1) acquisition and initial adjustments, (2) adding identifiers, and (3) constructing the final dataset. Files are saved with the “.ods” extension (compatible with LibreOffice Calc), and the **readODS** library is used to import them into the analysis workflow.

Guidelines

All files used in this workflow can be downloaded from our account on the Harvard Dataverse site:

Dataset

Python scripts (Jupyter notebooks) NAME

https://doi.org/10.7910/DVN/CBMKVA LINK

Dataset

Coordinates from sperm cells in wide format. NAME

https://doi.org/10.7910/DVN/JAN8RE LINK

Before start

Requirements

Two files are required. The first one should be the file with the coordinates of the trajectories (see Figure 1); the second file should contain the IDs of the analyzed sperm (see Figure 2). The second file can be generated from the file containing the motility parameters (see Figure 3). All necessary files can be downloaded from https://doi.org/10.7910/DVN/JAN8RE. In Figure 1 you can observe the data structure in the original ods file. It can be seen that the first column contains the sperm identifiers and the respective coordinate (*x* or *y*); the number corresponds to the identifier, and the *x* or *y* in parentheses indicates the value of the respective coordinate.

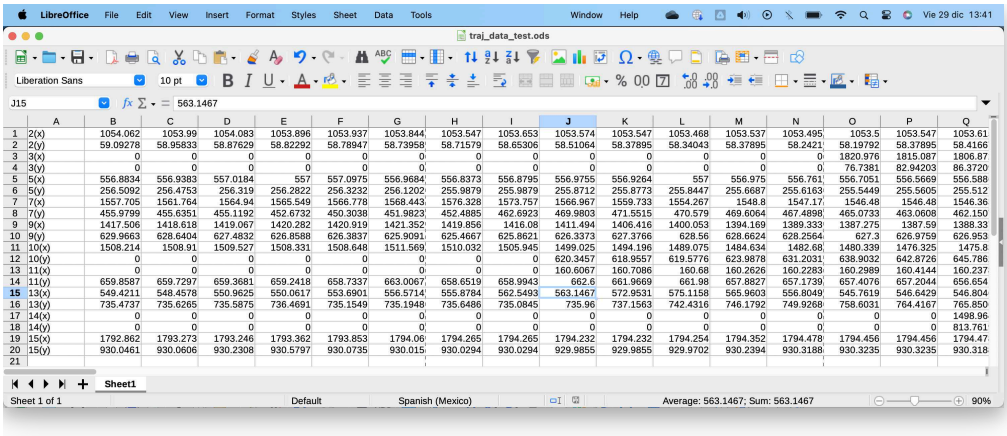


Figure 1. Screenshot of a LibreOffice Calc spreadsheet with coordinate data. The file traj_data_test.ods contains 20 rows and 150 columns. Each row corresponds to a sperm coordinate. The first column contains the sperm identifier, where the number is the indicator, and the letter in parentheses indicates the coordinate (*x* or *y*). The remaining 150 columns contain the coordinate value of each sperm in each frame analyzed by the CASA system. There may be rows where zeros are present instead of coordinate values; this is because some sperm may not have been detected throuhtout the capture routine.

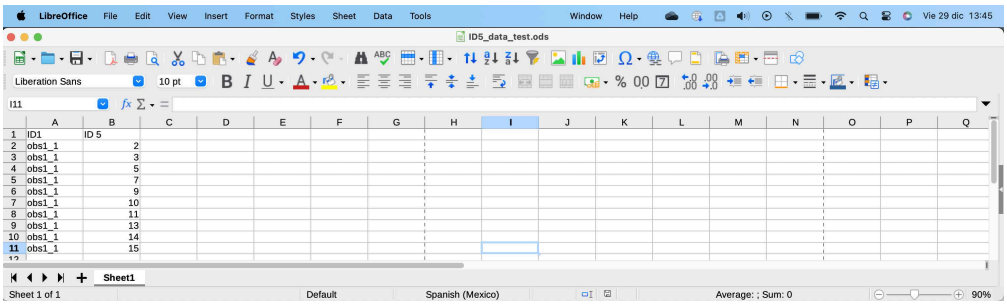
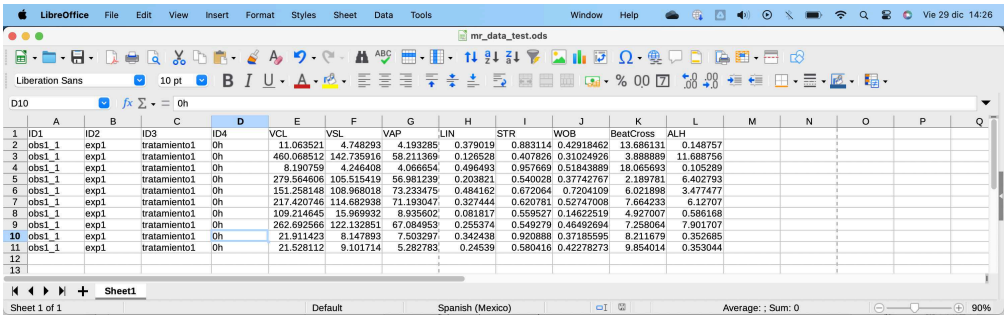


Figure 2. Screenshot of a LibreOffice Calc spreadsheet with sperm identifier data. The file has 11 rows, the firste row contains column names, and the remaining 10 rows have identifier data. The columns names correspond to two different identifiers, called ID1 and ID5.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	ID1	ID2	ID3	ID4	VCL	VSL	VAP	LIN	STR	WOB	BeatCross	ALH					
2	obs1_1	exp1	tratamiento1	0h	11.063521	4.748293	4.193285	0.379019	0.883114	0.42918462	13.686131	0.148757					
3	obs1_1	exp1	tratamiento1	0h	460.068512	142.735916	58.211369	0.126528	0.407826	0.31024926	3.888889	11.688756					
4	obs1_1	exp1	tratamiento1	0h	8.190759	4.246408	4.066654	0.496493	0.957669	0.51843889	18.065693	0.105289					
5	obs1_1	exp1	tratamiento1	0h	279.564606	105.515419	56.981239	0.203821	0.540028	0.37742767	2.189781	6.402793					
6	obs1_1	exp1	tratamiento1	0h	151.258148	108.968018	73.233475	0.484162	0.672064	0.7204109	6.021898	3.477477					
7	obs1_1	exp1	tratamiento1	0h	217.420746	114.682938	71.193047	0.327444	0.620781	0.52747008	7.664233	6.12707					
8	obs1_1	exp1	tratamiento1	0h	109.214645	15.969932	8.935602	0.081817	0.559527	0.14622519	4.927007	0.586168					
9	obs1_1	exp1	tratamiento1	0h	262.692566	122.132851	67.084953	0.255374	0.549279	0.46492694	7.258064	7.901707					
10	obs1_1	exp1	tratamiento1	0h	21.911423	8.147893	7.503297	0.342438	0.920888	0.37185595	8.211679	0.352685					
11	obs1_1	exp1	tratamiento1	0h	21.528112	9.101714	5.282783	0.24539	0.580416	0.42278273	9.854014	0.353044					
12																	
13																	

Figure 3. Screenshot of a LibreOffice Calc spreadsheet of the mr_data_test ods file. This file contains motility parameters data. The file comprises 11 rows; the first row contains the column names (motility parameters), and the subsequent 10 rows contain data for various evaluated motility parameters. The first four columns correspond to identifiers of the analyzed sperm.

Stage 1: Acquisition and Initial Adjustments

1 Load the necesary library for reading ods files and set the working directory

Command

new command name

```
library (readODS)
```

Command

new command name

```
setwd("/Users/andresammx/Documents/RStudio/Markdown/Coordenadas_esperm
aticas_formato")
```

1.1 Create an object named coord, and load into it the contents of the working file that contains the coordinates."

Command

new command name

```
coord<-read_ods("traj_data_test.ods", col_names=FALSE,
as_tibble=FALSE)
```

Note

The source file is read using the read_ods command. The argument col_names=FALSE specifies that the first line of the file should not be interpreted as column names. Similarly, the argument as_tibble=FALSE indicates the preference for obtaining an object with a dataframe structure rather than a tibble format.

Expected result

```
## New names:
## • `` -> `...1`
## • `` -> `...2`
## • `` -> `...3`
## • `` -> `...4`
## • `` -> `...5`
## • `` -> `...6`
## • `` -> `...7`
## • `` -> `...8`
## • `` -> `...9`
## • `` -> `...10`
## • `` -> `...11`
## • `` -> `...12`
## • `` -> `...13`
## • `` -> `...14`
## • `` -> `...15`
## • `` -> `...16`
## • `` -> `...17`
## • `` -> `...18`
## • `` -> `...19`
## • `` -> `...20`
## • `` -> `...21`
## • `` -> `...22`
## • `` -> `...23`
## • `` -> `...24`
## • `` -> `...25`
## • `` -> `...26`
## • `` -> `...27`
## • `` -> `...28`
## • `` -> `...29`
## • `` -> `...30`
## • `` -> `...31`
## • `` -> `...32`
## • `` -> `...33`
## • `` -> `...34`
## • `` -> `...35`
## • `` -> `...36`
## • `` -> `...37`
## • `` -> `...38`
## • `` -> `...39`
## • `` -> `...40`
## • `` -> `...41`
## • `` -> `...42`
## • `` -> `...43`
```

```
## • `` -> `...44`
## • `` -> `...45`
## • `` -> `...46`
## • `` -> `...47`
## • `` -> `...48`
## • `` -> `...49`
## • `` -> `...50`
## • `` -> `...51`
## • `` -> `...52`
## • `` -> `...53`
## • `` -> `...54`
## • `` -> `...55`
## • `` -> `...56`
## • `` -> `...57`
## • `` -> `...58`
## • `` -> `...59`
## • `` -> `...60`
## • `` -> `...61`
## • `` -> `...62`
## • `` -> `...63`
## • `` -> `...64`
## • `` -> `...65`
## • `` -> `...66`
## • `` -> `...67`
## • `` -> `...68`
## • `` -> `...69`
## • `` -> `...70`
## • `` -> `...71`
## • `` -> `...72`
## • `` -> `...73`
## • `` -> `...74`
## • `` -> `...75`
## • `` -> `...76`
## • `` -> `...77`
## • `` -> `...78`
## • `` -> `...79`
## • `` -> `...80`
## • `` -> `...81`
## • `` -> `...82`
## • `` -> `...83`
## • `` -> `...84`
## • `` -> `...85`
## • `` -> `...86`
## • `` -> `...87`
## • `` -> `...88`
```

```
## • `` -> `...89`
## • `` -> `...90`
## • `` -> `...91`
## • `` -> `...92`
## • `` -> `...93`
## • `` -> `...94`
## • `` -> `...95`
## • `` -> `...96`
## • `` -> `...97`
## • `` -> `...98`
## • `` -> `...99`
## • `` -> `...100`
## • `` -> `...101`
## • `` -> `...102`
## • `` -> `...103`
## • `` -> `...104`
## • `` -> `...105`
## • `` -> `...106`
## • `` -> `...107`
## • `` -> `...108`
## • `` -> `...109`
## • `` -> `...110`
## • `` -> `...111`
## • `` -> `...112`
## • `` -> `...113`
## • `` -> `...114`
## • `` -> `...115`
## • `` -> `...116`
## • `` -> `...117`
## • `` -> `...118`
## • `` -> `...119`
## • `` -> `...120`
## • `` -> `...121`
## • `` -> `...122`
## • `` -> `...123`
## • `` -> `...124`
## • `` -> `...125`
## • `` -> `...126`
## • `` -> `...127`
## • `` -> `...128`
## • `` -> `...129`
## • `` -> `...130`
## • `` -> `...131`
## • `` -> `...132`
## • `` -> `...133`
```

```
## • `` -> `...134`
## • `` -> `...135`
## • `` -> `...136`
## • `` -> `...137`
## • `` -> `...138`
## • `` -> `...139`
## • `` -> `...140`
## • `` -> `...141`
## • `` -> `...142`
## • `` -> `...143`
## • `` -> `...144`
## • `` -> `...145`
## • `` -> `...146`
## • `` -> `...147`
## • `` -> `...148`
## • `` -> `...149`
## • `` -> `...150`
## • `` -> `...151`
```

The output generated upon executing the function reveals that R has assigned numeric names to each column in the file.

1.2 Review the structure of the data in the *coord* object:

Command

new command name

```
str(coord)
```

Expected result

```
## 'data.frame':    20 obs. of  151 variables:
##  $ ...1  : chr  "2(x)" "2(y)" "3(x)" "3(y)" ...
##  $ ...2  : num  1054.1 59.1 0 0 556.9 ...
##  $ ...3  : num  1054 59 0 0 557 ...
##  $ ...4  : num  1054.1 58.9 0 0 557 ...
##  $ ...5  : num  1053.9 58.8 0 0 557 ...
##  $ ...6  : num  1053.9 58.8 0 0 557.1 ...
##  $ ...7  : num  1053.8 58.7 0 0 557 ...
##  $ ...8  : num  1053.5 58.7 0 0 556.8 ...
##  $ ...9  : num  1053.7 58.7 0 0 556.9 ...
##  $ ...10 : num  1053.6 58.5 0 0 557 ...
##  $ ...11 : num  1053.5 58.4 0 0 556.9 ...
##  $ ...12 : num  1053.5 58.3 0 0 557 ...
##  $ ...13 : num  1053.5 58.4 0 0 557 ...
##  $ ...14 : num  1053.5 58.2 0 0 556.8 ...
##  $ ...15 : num  1053.5 58.2 1821 76.7 556.7 ...
##  $ ...16 : num  1053.5 58.4 1815.1 82.9 556.6 ...
##  $ ...17 : num  1053.6 58.4 1806.9 86.4 556.6 ...
##  $ ...18 : num  1053.6 58.1 1798.5 82.3 556.8 ...
##  $ ...19 : num  1053.7 58.1 1793.4 79.3 556.6 ...
##  $ ...20 : num  1053.5 58 1789.5 76.6 556.6 ...
##  $ ...21 : num  1053.3 57.8 1787.1 74.7 556.6 ...
##  $ ...22 : num  1053.3 57.8 1787.2 82.1 556.5 ...
##  $ ...23 : num  1053.3 57.6 1793.8 82.6 556.6 ...
##  $ ...24 : num  1053.3 57.6 1800.7 79.9 556.7 ...
##  $ ...25 : num  1053 57.4 1805.8 76.5 556.5 ...
##  $ ...26 : num  1052.9 57.3 1807.2 72 556.4 ...
##  $ ...27 : num  1053.2 57.2 1807.7 67.4 556.5 ...
##  $ ...28 : num  1053.1 57.1 1807.5 63.1 556.7 ...
##  $ ...29 : num  1053.3 56.9 1806.7 59.6 556.6 ...
##  $ ...30 : num  1053 57 1807 56 557 ...
##  $ ...31 : num  1053.2 56.8 1807.8 54.2 556.6 ...
##  $ ...32 : num  1053.3 56.7 1807.1 54.2 556.5 ...
##  $ ...33 : num  1053 56.5 1807.2 52.3 556.5 ...
##  $ ...34 : num  1053.1 56.4 1811.1 42.5 556.5 ...
##  $ ...35 : num  1053.3 56.1 1822 56.3 556.6 ...
##  $ ...36 : num  1053.3 56.2 1832.9 70 556.5 ...
##  $ ...37 : num  1053.2 56.2 1843.8 83.8 556.4 ...
##  $ ...38 : num  1053.2 56 1835.8 89.5 556.4 ...
##  $ ...39 : num  1053.1 55.9 1830.9 93.3 556.4 ...
##  $ ...40 : num  1052.9 55.9 1822.9 94.7 556.4 ...
##  $ ...41 : num  1052.9 55.8 1815.2 93.8 556.4 ...
##  $ ...42 : num  1052.9 55.8 1809.9 92.3 556.4 ...
##  $ ...43 : num  1052.8 55.7 1808.7 91.9 556.4 ...
```

```
##  $ ...44 : num  1052.4 55.7 1815.9 93 556.4 ...
##  $ ...45 : num  1052.4 55.6 1823.7 92.9 556.3 ...
##  $ ...46 : num  1052.4 55.5 1828.1 90 556.5 ...
##  $ ...47 : num  1052.4 55.5 1830 86 556.4 ...
##  $ ...48 : num  1052.5 55.2 1830.1 81.8 556.4 ...
##  $ ...49 : num  1052.5 55 1829.9 77.1 556.5 ...
##  $ ...50 : num  1052.3 55 1830 72.8 556.5 ...
##  $ ...51 : num  1052.3 54.9 1828.6 70.4 556.5 ...
##  $ ...52 : num  1052.4 54.7 1827.7 68.7 556.4 ...
##  $ ...53 : num  1052.3 54.7 1827.5 68.7 556.4 ...
##  $ ...54 : num  1052.1 54.7 1827 66.1 556.3 ...
##  $ ...55 : num  1052.1 54.7 1828.4 57.1 556.4 ...
##  $ ...56 : num  1051.9 54.6 1836.3 66.6 556.4 ...
##  $ ...57 : num  1051.9 54.3 1844.1 76.2 556.2 ...
##  $ ...58 : num  1052.1 54.2 1852 85.8 556.4 ...
##  $ ...59 : num  1051.9 54.1 1859.9 95.4 556.2 ...
##  $ ...60 : num  1052 54.1 1855.6 100.8 556.2 ...
##  $ ...61 : num  1052 53.9 1849.1 103.1 556.1 ...
##  $ ...62 : num  1052 53.8 1839.2 102.1 556.2 ...
##  $ ...63 : num  1052 53.7 1831.6 100.5 556 ...
##  $ ...64 : num  1052 53.8 1829.4 100.6 556.1 ...
##  $ ...65 : num  1051.9 53.8 1833.8 102.7 556 ...
##  $ ...66 : num  1052 53.8 1842.8 103.5 556.1 ...
##  $ ...67 : num  1051.7 53.6 1848.3 100.6 556.2 ...
##  $ ...68 : num  1051.5 53.4 1851.2 96.3 556.3 ...
##  $ ...69 : num  1051.7 53.4 1851.6 92 556.2 ...
##  $ ...70 : num  1051.6 53.6 1849.9 87.7 556.1 ...
##  $ ...71 : num  1051.5 53.2 1847.8 85 556.1 ...
##  $ ...72 : num  1051.7 53.1 1846.1 81.8 556 ...
##  $ ...73 : num  1051.7 53 1844.2 79.1 555.8 ...
##  $ ...74 : num  1051.9 53 1843.1 76.6 555.8 ...
##  $ ...75 : num  1052 53 1844.3 74.2 555.9 ...
##  $ ...76 : num  1051.9 53.1 1843.9 64.3 555.7 ...
##  $ ...77 : num  1051.8 53 1853.2 73.9 555.9 ...
##  $ ...78 : num  1051.9 52.9 1862.6 83.5 555.9 ...
##  $ ...79 : num  1051.9 52.8 1871.9 93.1 555.9 ...
##  $ ...80 : num  1051.9 52.8 1881.3 102.7 555.9 ...
##  $ ...81 : num  1051.9 52.8 1873 105.4 555.8 ...
##  $ ...82 : num  1051.9 52.8 1865.4 107.8 555.8 ...
##  $ ...83 : num  1051.9 52.8 1855.8 106.4 555.8 ...
##  $ ...84 : num  1051.7 52.8 1848.2 104 555.7 ...
##  $ ...85 : num  1051.6 52.8 1845.2 102.6 555.8 ...
##  $ ...86 : num  1051.6 52.8 1848 104.9 555.7 ...
##  $ ...87 : num  1051.7 52.8 1857.1 106.3 555.8 ...
##  $ ...88 : num  1051.7 52.8 1861.8 104.1 555.7 ...
```

```
##  $ ...89 : num  1051.9 52.9 1864.8 100.5 555.7 ...
##  $ ...90 : num  1051.9 52.8 1866 96.7 555.8 ...
##  $ ...91 : num  1051.9 52.9 1865.7 93.2 555.8 ...
##  $ ...92 : num  1051.5 52.9 1864.1 90.7 555.7 ...
##  $ ...93 : num  1051.5 52.8 1862.9 87.9 555.7 ...
##  $ ...94 : num  1051.5 52.7 1861.6 85.3 555.6 ...
##  $ ...95 : num  1051.6 52.8 1861.3 83.4 555.5 ...
##  $ ...96 : num  1051.3 52.7 1860.7 82.7 555.6 ...
##  $ ...97 : num  1051.4 52.7 1860.2 81.3 555.6 ...
##  $ ...98 : num  1051.4 52.6 1857.6 68.8 555.6 ...
##  $ ...99 : num  1051.4 52.6 1860 84.7 555.7 ...
##   [list output truncated]
```

Note

The coord object contains 20 observations (rows) and 151 variables (columns). Specifically, the coord object has 151 columns, where the first column represents sperm identifiers, and the remaining columns correspond to the values of the *x* and *y* coordinates.

1.3 Create a new object named coord2. This step involves transposing rows into columns.

Command

new command name

```
coord2<-t(sapply(coord, as.numeric))
```

Note

The t function is applied to each element of coord using the sapply function. An argument is provided to sapply to ensure that the result is returned as numeric."

Expected result

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by
coercion
```

A warning message is generated, indicating that the function has introduced NA values.

1.4 Assign the structure of a dataframe to the coord2 object

Command

new command name

```
coord2<-as.data.frame(coord2)
```

1.5 Verify the structure of coord2 object

Command

new command name

```
str(coord2)
```

Expected result

```
## 'data.frame':    151 obs. of  20 variables:
##  $ V1 : num  NA 1054 1054 1054 1054 ...
##  $ V2 : num  NA 59.1 59 58.9 58.8 ...
##  $ V3 : num  NA 0 0 0 0 0 0 0 0 0 ...
##  $ V4 : num  NA 0 0 0 0 0 0 0 0 0 ...
##  $ V5 : num  NA 557 557 557 557 ...
##  $ V6 : num  NA 257 256 256 256 ...
##  $ V7 : num  NA 1558 1562 1565 1566 ...
##  $ V8 : num  NA 456 456 455 453 ...
##  $ V9 : num  NA 1418 1419 1419 1420 ...
##  $ V10: num  NA 630 629 627 627 ...
##  $ V11: num  NA 1508 1509 1510 1508 ...
##  $ V12: num  NA 0 0 0 0 ...
##  $ V13: num  NA 0 0 0 0 ...
##  $ V14: num  NA 660 660 659 659 ...
##  $ V15: num  NA 549 548 551 550 ...
##  $ V16: num  NA 735 736 736 736 ...
##  $ V17: num  NA 0 0 0 0 0 0 0 0 0 ...
##  $ V18: num  NA 0 0 0 0 0 0 0 0 0 ...
##  $ V19: num  NA 1793 1793 1793 1793 ...
##  $ V20: num  NA 930 930 930 931 ...
```

Note

We have a dataframe with 151 observations (rows) and 20 variables (columns), where the first row contains NA values.

1.6 Examine the beginning of the coord2 object to verify its structure.

Command

new command name

```
head(coord2)
```

Expected result

```
##            V1       V2 V3 V4       V5       V6       V7
V8       V9
## ...1       NA       NA NA NA       NA       NA       NA
NA       NA
## ...2 1054.062 59.09278  0  0 556.8834 256.5092 1557.705
455.9799 1417.506
## ...3 1053.990 58.95833  0  0 556.9383 256.4753 1561.764
455.6351 1418.618
## ...4 1054.083 58.87629  0  0 557.0184 256.3190 1564.940
455.1192 1419.067
## ...5 1053.896 58.82292  0  0 557.0000 256.2822 1565.549
452.6732 1420.282
## ...6 1053.937 58.78947  0  0 557.0975 256.3232 1566.778
450.3038 1420.919
##           V10      V11 V12 V13      V14      V15      V16
V17 V18      V19
## ...1       NA       NA  NA  NA       NA       NA       NA
NA  NA       NA
## ...2 629.9663 1508.214   0   0 659.8587 549.4211 735.4737
0   0 1792.862
## ...3 628.6404 1508.910   0   0 659.7297 548.4578 735.6265
0   0 1793.273
## ...4 627.4832 1509.527   0   0 659.3681 550.9625 735.5875
0   0 1793.246
## ...5 626.8588 1508.331   0   0 659.2418 550.0617 736.4691
0   0 1793.362
## ...6 626.3837 1508.648   0   0 658.7337 553.6901 735.1549
0   0 1793.853
##           V20
## ...1       NA
## ...2 930.0461
## ...3 930.0606
## ...4 930.2308
## ...5 930.5797
## ...6 930.0735
```

Note

At the beginning of the coord2 object, a row containing NA values has been inserted. To clean the object, select only the rows that are required.

1.7 Select only the rows that are required, and then review the beginning of the coord2 object again.

Command

new command name

```
coord2<-coord2[2:151,]
head(coord2)
```

Expected result

```
##            V1       V2 V3 V4       V5       V6       V7
V8       V9
## ...2 1054.062 59.09278  0  0 556.8834 256.5092 1557.705
455.9799 1417.506
## ...3 1053.990 58.95833  0  0 556.9383 256.4753 1561.764
455.6351 1418.618
## ...4 1054.083 58.87629  0  0 557.0184 256.3190 1564.940
455.1192 1419.067
## ...5 1053.896 58.82292  0  0 557.0000 256.2822 1565.549
452.6732 1420.282
## ...6 1053.937 58.78947  0  0 557.0975 256.3232 1566.778
450.3038 1420.919
## ...7 1053.844 58.73958  0  0 556.9684 256.1202 1568.443
451.9823 1421.352
##           V10      V11 V12 V13      V14      V15      V16
V17 V18      V19
## ...2 629.9663 1508.214   0   0 659.8587 549.4211 735.4737
0   0 1792.862
## ...3 628.6404 1508.910   0   0 659.7297 548.4578 735.6265
0   0 1793.273
## ...4 627.4832 1509.527   0   0 659.3681 550.9625 735.5875
0   0 1793.246
## ...5 626.8588 1508.331   0   0 659.2418 550.0617 736.4691
0   0 1793.362
## ...6 626.3837 1508.648   0   0 658.7337 553.6901 735.1549
0   0 1793.853
## ...7 625.9091 1511.569   0   0 663.0067 556.5714 735.1948
0   0 1794.060
##           V20
## ...2 930.0461
## ...3 930.0606
## ...4 930.2308
## ...5 930.5797
## ...6 930.0735
## ...7 930.0150
```

In the coord2 object, the odd-numbered columns correspond to x-coordinates, while the even-numbered columns correspond to y-coordinates. Although the order has

been maintained, the sperm identifiers have been lost.

1.8 Reshape the coord2 object into a long format. Place column 3 below column 1, column 5 below column 3, and so on. Apply the same restructuring to the even-numbered columns. Begin this process by creating an object named col_odd, which will match the length of coord2 and store the identifiers for the odd-numbered columns.

Command

new command name

```
col_odd<-seq_len(ncol(coord2)) %% 2
```

Next, we will create two objects named *only_x* and *only_y*, where we will place the odd-numbered columns (corresponding to *x*) and even-numbered columns (corresponding to *y*), respectively:"

Command

new command name

```
only_x<-coord2[, col_odd == 1]
only_y<-coord2[, col_odd == 0]
```

1.9 Create two objects named only_x and only_y. Place the odd-numbered columns (corresponding to x) in only_x and the even-numbered columns (corresponding to y) in only_y.

Command

new command name

```
only_x<-coord2[, col_odd == 1]
only_y<-coord2[, col_odd == 0]
```

1.10 Verify the content of each object:

Command

new command name

```
head(only_x)
```

Expected result

```
##            V1 V3       V5       V7       V9      V11 V13
V15 V17      V19
## ...2 1054.062  0 556.8834 1557.705 1417.506 1508.214   0
549.4211   0 1792.862
## ...3 1053.990  0 556.9383 1561.764 1418.618 1508.910   0
548.4578   0 1793.273
## ...4 1054.083  0 557.0184 1564.940 1419.067 1509.527   0
550.9625   0 1793.246
## ...5 1053.896  0 557.0000 1565.549 1420.282 1508.331   0
550.0617   0 1793.362
## ...6 1053.937  0 557.0975 1566.778 1420.919 1508.648   0
553.6901   0 1793.853
## ...7 1053.844  0 556.9684 1568.443 1421.352 1511.569   0
556.5714   0 1794.060
```

Command

new command name

```
head(only_y)
```

Expected result

```
##            V2 V4       V6       V8      V10 V12      V14
V16 V18      V20
## ...2 59.09278  0 256.5092 455.9799 629.9663   0 659.8587
735.4737   0 930.0461
## ...3 58.95833  0 256.4753 455.6351 628.6404   0 659.7297
735.6265   0 930.0606
## ...4 58.87629  0 256.3190 455.1192 627.4832   0 659.3681
735.5875   0 930.2308
## ...5 58.82292  0 256.2822 452.6732 626.8588   0 659.2418
736.4691   0 930.5797
## ...6 58.78947  0 256.3232 450.3038 626.3837   0 658.7337
735.1549   0 930.0735
## ...7 58.73958  0 256.1202 451.9823 625.9091   0 663.0067
735.1948   0 930.0150
```

1.11 Create two new objects to contain the stacked data from the x-columns and y-columns, respectively:

Command

new command name

```
only_x<-data.frame(x=unlist(only_x, use.names=FALSE))
only_y<-data.frame(y=unlist(only_y, use.names=FALSE))
```

1.12 Create a new object named traj with two columns: the first column will contain the data from the only_x object, and the second column will contain the data from the only_y object."

Command

new command name

```
traj<-cbind(only_x, only_y)
```

1.13 Verify the structure of the *traj* object

Command

new command name

```
str(traj)
```

Expected result

```
## 'data.frame':    1500 obs. of  2 variables:
##  $ x: num  1054 1054 1054 1054 1054 ...
##  $ y: num  59.1 59 58.9 58.8 58.8 ...
```

The traj object now contains two variables (the x and y columns) and 1,500 observations (or rows) of data. In total, the dataframe holds 3,000 data points, calculated by multiplying the 150 pairs of coordinates by 10 observations (sperm) per 2 coordinates."

Stage 2: Identifiers creation

2 Add columns with string-type identifiers for each sperm to the traj object.
ID1: Key for the capture routine
ID2: Experiment number (or the identifier of the male or experimental unit)
ID3: Experimental treatment (or factor level)
ID4: Incubation time
ID5: Sperm identifier in the capture routine

Note

Each identifier must match exactly with the corresponding entries in the motility parameters file. It is important to note that the content of these identifying columns will vary depending on the input files.

Create a file containing the coordinates for the entire experiment. Prepare a file specifying the order of IDs for the analyzed sperm with care. Import this file into R, and extract the ID1 and ID5 objects from it.
Finally, create an object to include the missing identifiers. This object should contain the data presented in Figure 2.

Command

new command name

```
ID_faltantes<-read_ods("ID5_data_test.ods", col_names=TRUE,
as_tibble=FALSE)
```

2.1 Verify the structure of the *ID_faltantes* object

Command

new command name

```
str(ID_faltantes)
```

Expected result

```
## 'data.frame':    10 obs. of  2 variables:
##  $ ID1 : chr  "obs1_1" "obs1_1" "obs1_1" "obs1_1" ...
##  $ ID.5: num  2 3 5 7 9 10 11 13 14 15
```

Since the traj object contains 150 coordinates for each sperm, ensure that each identifier is repeated 150 times.

2.2 Extract column 1 from the ID_faltantes object and assign it to a new object named ID1. Then, create 150 repetitions of each value in ID1.

Command

new command name

```
ID1<-ID_faltantes[,1]
ID1<-rep(ID1, each=150)
length(ID1)
```

Expected result

```
## [1] 1500
```

2.3 Create an object named num_row to account for the varying number of lines (analyzed sperm) in each input file. This object should contain the required number of lines specific to the input file being processed.

Command

new command name

```
num_row<-(ncol(coord2)/2)*150
num_row
```

Expected result

[1] 1500

2.4 Use the num_row object to generate the required number of lines for ID2, ID3, and ID4.

Command

new command name

```
ID2<-rep("exp1", num_row)
ID3<-rep("tratamiento1", num_row)
ID4<-rep("0h", num_row)
```

2.5 Create the ID5 object by extracting the content of column 2 from the ID_faltantes object and repeating it 150 times.

Command

new command name

```
ID5<-ID_faltantes[,2]
ID5<-rep(ID5, each=150)
```

Stage 3: Final object creation

3 Finally, create a new object named traj2 by merging the identifying columns with the traj object. Ensure to account for the fact that the sperm identifiers vary throughout the dataframe.

Command

new command name

```
traj2<-as.data.frame(cbind(ID1, ID2, ID3, ID4, ID5, traj))
str(traj2)
```

Expected result

```
## 'data.frame':    1500 obs. of  7 variables:
##  $ ID1: chr  "obs1_1" "obs1_1" "obs1_1" "obs1_1" ...
##  $ ID2: chr  "exp1" "exp1" "exp1" "exp1" ...
##  $ ID3: chr  "tratamiento1" "tratamiento1" "tratamiento1" "tratamiento1" ...
##  $ ID4: chr  "0h" "0h" "0h" "0h" ...
##  $ ID5: num  2 2 2 2 2 2 2 2 2 2 ...
##  $ x  : num  1054 1054 1054 1054 1054 ...
##  $ y  : num  59.1 59 58.9 58.8 58.8 ...
```

3.1 Review both the beginning and the end of the traj2 object to verify that the data has been merged correctly.

Command

new command name

```
head(traj2)
```

Expected result

```
##     ID1  ID2         ID3 ID4 ID5       x       y
## 1 obs1_1 exp1 tratamiento1  0h   2 1054.062 59.09278
## 2 obs1_1 exp1 tratamiento1  0h   2 1053.990 58.95833
## 3 obs1_1 exp1 tratamiento1  0h   2 1054.083 58.87629
## 4 obs1_1 exp1 tratamiento1  0h   2 1053.896 58.82292
## 5 obs1_1 exp1 tratamiento1  0h   2 1053.937 58.78947
## 6 obs1_1 exp1 tratamiento1  0h   2 1053.844 58.73958
```

Command

new command name

```
tail(traj2)
```

Expected result

```
##       ID1  ID2        ID3 ID4 ID5       x       y
## 1495 obs1_1 exp1 tratamiento1  0h  15 1805.636 931.2121
## 1496 obs1_1 exp1 tratamiento1  0h  15 1805.554 931.2769
## 1497 obs1_1 exp1 tratamiento1  0h  15 1805.403 931.4478
## 1498 obs1_1 exp1 tratamiento1  0h  15 1804.460 930.5397
## 1499 obs1_1 exp1 tratamiento1  0h  15 1804.113 931.2903
## 1500 obs1_1 exp1 tratamiento1  0h  15 1803.900 931.3500
```

3.2 Check for undetected sperm in the 150 frames due to CASA system settings, as this may result in zero values. These zero values can cause issues when reconstructing trajectory images and must be eliminated. To address this, follow two steps: first, replace all zero values with NA. Then, use the drop_na function from the tidyr library to remove all rows containing NA values.

Command

new command name

```
traj2[traj2 == 0]<-NA
```

Command

new command name

```
library(tidyr)
traj2<- drop_na(traj2)
```

3.3 Verify the traj2 object

Command

new command name

```
str(traj2)
```

Expected result

```
## 'data.frame':    1379 obs. of  7 variables:
##  $ ID1: chr  "obs1_1" "obs1_1" "obs1_1" "obs1_1" ...
##  $ ID2: chr  "exp1" "exp1" "exp1" "exp1" ...
##  $ ID3: chr  "tratamiento1" "tratamiento1" "tratamiento1" "tratamiento1" ...
##  $ ID4: chr  "0h" "0h" "0h" "0h" ...
##  $ ID5: num  2 2 2 2 2 2 2 2 2 2 ...
##  $ x  : num  1054 1054 1054 1054 1054 ...
##  $ y  : num  59.1 59 58.9 58.8 58.8 ...
```

3.4 Verify that all rows containing zeros have been successfully eliminated.

Command

new command name

```
head(traj2)
```

Expected result

```
##     ID1  ID2        ID3 ID4 ID5       x       y
## 1 obs1_1 exp1 tratamiento1  0h   2 1054.062 59.09278
## 2 obs1_1 exp1 tratamiento1  0h   2 1053.990 58.95833
## 3 obs1_1 exp1 tratamiento1  0h   2 1054.083 58.87629
## 4 obs1_1 exp1 tratamiento1  0h   2 1053.896 58.82292
## 5 obs1_1 exp1 tratamiento1  0h   2 1053.937 58.78947
## 6 obs1_1 exp1 tratamiento1  0h   2 1053.844 58.73958
```

3.5 Verify that the sperm IDs in the traj2 object match the IDs of the rows in the motility parameters file.

Command

new command name

```
unique(traj2$ID5)
```

Expected result

```
## [1]  2  3  5  7  9 10 11 13 14 15
```

3.6 Create a CSV file at this point for future use.

Command

new command name

```
write.csv(traj2, "traj_exp1.csv")
```

Alternatively, create a new object with a different name, such as:

Command

new command name

```
end_1<-traj2
```

3.7 If repeating the workflow with other files, use the last created object (end_1) to generate a final object containing all the data from the processed files. Proceed as follows:

Command

new command name

```
end_all<-as.data.frame(rbind(end_1, end_2, end_3, end_4, end_5,
end_6, end_7, end_8))
```

Note

When processing one or more files with trajectory data, ensure that the traj file and the motility parameters file have the same number of observations (lines) to successfully combine them in the traj-ah-6-full-0.ipynb Jupyter notebook (see Figure 3).

Note

It is also important to verify that the input files intended for use in the traj-ah-6-full-0.ipynb notebook have a CSV extension. For the coordinate file, remove the first line containing the column names. In contrast, ensure that the motility parameters file includes the line with the column names.

Conclusions

4 With the proposed workflow in this document, one can modify a coordinate file to obtain the long format. Similarly, sperm IDs can be automatically generated so that each set of coordinates has its own identifier.Using the proposed workflow in this document, modify a coordinate file to obtain the long format. Additionally, automatically generate sperm IDs to ensure that each set of coordinates has its own unique identifier.

Protocol references

Amann RP and Waberski D (2014) Computer-assisted sperm analysis (CASA): Capabilities and potential developments. *Theriogenology* **81** 5-17.e1-3.
Giaretta E, Munerato M, Yeste M, Galeati G, Spinaci M, Tamanini C, Mari G and Bucci D (2017) Implementing an open-access CASA software for the assessment of stallion sperm motility: Relationship with other sperm quality parameters. *Animal Reproduction Science* **176** 11–19.
Ramón M and Martínez-Pastor F (2018) Implementation of novel statistical procedures and other advanced approaches to improve analysis of CASA data. *Reproduction, Fertility, and Development* **30** 860–866.
Rasband WS (1997) ImageJ, US National Institutes of Health. Bethesda, Maryland, USA.
Rivas AC, Ayala EME and Aragon MA (2022) Effect of various pH levels on the sperm kinematic parameters of boars. *South African Journal of Animal Science* **52** 693–704.
Rodríguez-Martínez EA, Rivas CU, Ayala ME, Blanco-Rodríguez R, Juarez N, Hernandez-Vargas EA and Aragón A (2023) A new computational approach, based on images trajectories, to identify the subjacent heterogeneity of sperm to the effects of ketanserin. *Cytometry. Part A* **103** 655–663.
Wilson-Leedy JG and Ingermann R (2007) Development of a novel CASA system based on open source software for characterization of zebrafish sperm motility parameters. *Theriogenology* **67** 661–672.

Acknowledgements

This work was supported by UNAM-PAPIIT IN224925.
This work was supported by UNAM-PAPIIT IN215425.
This work was supported by the Programa de Becas Posdoctorales de la Direction General de Asuntos del Personal Académico (DGAPA), UNAM to the postdoctoral fellow Cindy Ursula Rivas Arzaluz.
The authors express their gratitude to Dr. Masakatsu Fujinoki for sharing his data.