

¹
Draft v0.1

²
Benjamin Gallois

³
February 5, 2021

Contents

4	I FastTrack	6
5	1 Introduction	7
6	1.1 Video based tracking	7
7	1.2 The tracking, a not so simple problem	9
8	1.3 Existing software	10
9	1.3.1 Proprietary software	10
10	1.3.2 Open-source software	12
11	1.4 FastTrack: an original approach	17
12	2 Design and implementation	19
13	2.1 Tools used	19
14	2.2 Implementation	20
15	2.2.1 Detection	21
16	2.2.2 Association	26
17	2.2.3 Automatic tracking parameters	28
18		

19	2.2.4	Manual correction	30
20	2.2.5	Analysis	32
21	2.3	Deployment	33
22	2.3.1	CI/CD	33
23	2.3.2	Documentation	35
24	3	Movies dataset	36
25	4	Results	39
26	4.1	Performance	39
27	4.2	Dataset classification	42
28	4.3	Parameters optimization	46
29	5	Perspective	50
30	Appendices		52
31	A	Voronoi diagram	53
32	A.1	Definition	53
33	A.2	Construction	53
34	B	Hungarian algorithm	55
35	B.1	Definition	55
36	B.2	Description	55

37	C Displacement distribution fitting	58
38	II Dual	59
39	6 Introduction	60
40	6.1 The chemical perception	60
41	6.1.1 Olfaction	62
42	6.1.2 Gustation	65
43	6.1.3 Common chemical sense	67
44	6.1.4 Behavior	67
45	6.2 Behavioral studies	69
46	6.2.1 Conditioned place preference	69
47	6.2.2 Well-plate	70
48	6.2.3 Diffusion	72
49	6.2.4 Flow	73
50	7 Experimental setups	77
51	7.1 Dual	77
52	7.1.1 Overview	77
53	7.1.2 Construction	78
54	7.1.3 Construction and usage	82

55	7.2 The Tropical River	83
56	7.2.1 Description	84
57	7.2.2 Usage and limitation	86
58	8 Results	88
59	8.1 Methods	88
60	8.1.1 Experiment	88
61	8.1.2 Analysis	90
62	8.2 Results	95
63	8.2.1 Setup caracterisation	95
64	8.2.2 Products screening	98
65	9 Conclusion	107
66	Appendices	110
67	D Visualisation	111
68	D.1 Boxplot	111
69	D.2 Violinplot	111
70	E Calculation	113
71	E.1 Mean quantities	113
72	E.2 Statistical tests	113

73	E.3 Image analysis	114
74	F Markov Model	115
75	G Preference index distributions	117

Part I

76

FastTrack

77

⁷⁸ **Chapter 1**

⁷⁹ **Introduction**

Talk is cheap. Show me the code.

Linux Torvald

⁸⁰ **1.1 Video based tracking**

The tracking of objects from video recordings is a problem that has gained much popularity in recent years. It is mostly due to its great potential, both in academics and for commercial and security applications. Examples include autonomous cars that can drive themselves, or the Airbus ATTOL project [1] that allows fully automated take-off, landing, and taxiing of plane based solely on image analysis. A large part of the research effort is focused on creating pedestrian recognition and tracking algorithms to automate the analysis of video surveillance data. Tracking is also widely used in movie creation with special effects (VFX, CGI), whether to stabilize shots or to realize special effects (e.g., motion capture), and for industrial process [2]. In this case, automated tracking on images re-

92 duces costs, production time, and human operators' use.

93 In academics, the use of automated tracking, especially in biology and
94 ecology [3, 4], is a rapidly growing field because it avoids the need to dis-
95 turb animals by invasively mark them. Cellular motions tracking is also
96 widely studied with very specialized algorithms developed solely for this
97 purpose [5, 6]. Other fields of science are interested in automated track-
98 ing, to name a few: microfluidic [7], active matter [8], social science [9]
99 and robotic [10]. It allows generating a large amount of reliable data, re-
100 ducing bias, and avoiding a long and laborious manual analysis (that is
101 sometimes impossible).

102 Object tracking can be separated into two categories: the Single Ob-
103 ject Tracking (SOT), where the goal is to detect a single object in a more
104 or less complicated scene, and the Multiple Object Tracking (MOT), where
105 the goal is to detect and track several objects. In this dissertation, we will
106 place ourselves within the MOT framework because it regroups the vast
107 majority of scientific applications. In a scientific experiment, the track-
108 ing problem's difficulty is reduced by designing the setup well. In general,
109 the image quality is good, the camera is fixed, and we can optimize the
110 lighting to facilitate object detection. On the other hand, the tolerance to
111 errors is low if one wants to produce reliable data and robust scientific
112 conclusions. A decisive point is the algorithm's performance, which must
113 analyze the data in a reasonable time compared to their production rate
114 and meet the user's material and technical constraints. The ease of instal-
115 lation and use of the software that integrates the algorithm should not
116 be neglected. The users brought to use these software are generally not
117 experts in computer science and image analysis, and the software must
118 be readily installable and usable by all.

119 We will first see how tracking is a complex problem and how we can re-
120 duce or bypass this complexity. We will then present a non-exhaustive list
121 of existing tracking software applied to diverse scientific fields. Finally, we
122 will see how FastTrack's approach, the software we have created to solve
123 the tracking problem, is different, and in which cases it can be useful.

1.2 The tracking, a not so simple problem

The image-based tracking of objects rest on three key steps: the acquisition of the images, which, depending on the acquisition parameters, will condition the difficulty of the tracking and the type of algorithm that can be used; the detection of objects, which consists in separating the objects from the background; and finally the assignment of objects from one image to another allowing to keep track of the objects' identities. Object tracking is generally a complex image processing task [3]. Depending on the objects studied, each step can be difficult. For example, animals are highly deformable objects interacting with each other, making the detection step complex. The scene can be complicated, with objects disappearing behind the decor elements, superimposing each other (the so-called occlusion phenomenon), or entering and leaving the field of view, complicating the detection and the association step.

Object detection problems can usually be circumvented by the design of the experimental setup whenever it is possible. A fixed point of view and lighting optimization usually allow simple detection by subtracting a background image (without objects) and applying a threshold. For more complicated cases, a wide variety of algorithms are available [11] and applicable depending on the images' quality. The most common is to detect points of interest in the object. This technique is invariant to the point of view and illumination but requires a good image quality. Segmentation allows to separate the image by area of similarities and thus to detect objects of interest, many algorithms and approaches exist to segment an image. Machine learning can also be applied for objects detection [12].

Two main classes of algorithms can be distinguished to mitigate association problems. The first class of algorithms uses the object's kinematic quantities, such as direction or position [13], to predict or find the position of the object on the next image and thus keep its identity. This method's error rate remains constant when we increase the number of individuals (keeping the density of objects fixed). It is generally fast, and

155 this makes it a good candidate for real-time tracking applications. The
156 major disadvantage of this approach comes from the error propagation
157 phenomenon. If the algorithm makes an error in the assignment, it has no
158 way to correct the error at the next step, and it propagates to the end of
159 the analysis. The second class of algorithms is based on recognizing the
160 object's points of interest, allowing the creation of a "fingerprint" unique
161 to each object. That can be done using either a classical method [14, 15],
162 or using machine learning [16, 17]. This technique solves the propagation
163 of errors problem and allows objects to be tracked over time, i.e., across
164 several unpaired videos. For example, an animal can be recognized from
165 one day of experiments to the next, which can be very useful, especially
166 for behavioral studies. This method requires an image of sufficient qual-
167 ity to extract markers representative of the object. It also requires more
168 computational resources, thus, an analysis that cannot be done in real-
169 time. However, the main limitation is the number of objects it can track. It
170 is currently limited to about ten objects per image with classical methods
171 before the algorithms' performance degrades significantly. The machine
172 learning approach makes it possible to increase the number of objects at
173 the cost of long computation time and the need to use high-performance
174 computers.

175 **1.3 Existing software**

176 Many tracking software already exist. We will make a non-exhaustive
177 list of the most popular ones, separating them into two categories: pro-
178 prietary software and open-source software.

179 **1.3.1 Proprietary software**

180 The proprietary software presented here are closed-source. The user
181 cannot modify the code to adapt the software to his project or check

182 precisely how the tracking is performed. On the other hand, they do not
183 require any computer knowledge and benefit from a support service con-
184 venient for users that do not have a lot of computer knowledge. They are
185 an excellent alternative to other options that are sometimes difficult to
186 implement, but their high price can be a hindrance for some users.

187 **EthoVision XT** EthoVision XT is a software developed by the company
188 Noldus. It accompanies the user from the acquisition of images, thanks to
189 a system of experiment templates, to the data analysis with a module al-
190 lowing to visualize standard behavioral parameters. The software is com-
191 plete and widely used. It is somewhat specialized in the field of behavioral
192 neurosciences. It includes modules for classical behavioral experiments
193 (e.g., water-maze, rats social interaction). It also allows performing live
194 tracking so that users do not have to save images for long experiments.

195 EthoVision XT is a mature software. A large number of modules are
196 available as well as a system that allows the user to create its own exper-
197 iment template. The most significant disadvantage is that the user cannot
198 modify the software or control how the tracking is done. Price can be a
199 barrier for some users, as the software costs a minimum of 5,850 USD
200 without modules, and it is compatible only with Windows. Focused on
201 tracking animals, it will not be suitable for other systems.

202 **Any-maze** Any-maze is a software developed by Stoelting Co. It is spe-
203 cialized in the analysis of behavioral neuroscience experiments. It directly
204 integrates tools for standardized tests (e.g., forced swim test, fear condi-
205 tioning test), allowing fully automated analysis of these experiments. It
206 can track in real-time or from recorded videos.

207 Any-maze is a complete solution for creating and analyzing typical be-
208 havioral experiments. It can be purchased with the experimental setup
209 already optimized and calibrated for the software. The Any-maze suite
210 consists of three software packages. The tracking part is available for USD

211 6,495 or USD 1,500 per year. The software is available for Windows only.

212 **ToxTrack** ToxTrack [18] is a software that implements in a graphical in-
213 terface the ToxId algorithm [19]. To summarize, the algorithm extracts ob-
214 jects from the background by applying a threshold. The pieces of trajec-
215 tories between occlusions are divided into short and long trajectories based
216 on a user-defined threshold time. A group of long trajectories where all
217 individuals are observed simultaneously is then extracted. In this case,
218 the assignment is made using the Hungarian algorithm Figure ???. The re-
219 maining trajectories are then assigned to the corresponding object se-
220 lecting the best correlation value in a trajectory identification matrix Fig-
221 ure 1.1. This matrix contains the similarity between every two trajectory
222 fragments based on objects' features. The authors report that ToxId is as
223 powerful as existing software, speedy, and can track objects in real-time.
224 A disadvantage that can be seen in this algorithm is that it only works
225 for a constant number of animals. The algorithm's initialization requires
226 to have at one moment t all the objects to be tracked simultaneously
227 detectable for a user-defined time $t + dt$. The user-interface (UI) is some-
228 times difficult to use: the integrated tracking reviewer does not permit to
229 correct the tracking or to replay the tracking frame by frame.

230 The UI includes tools to define areas of interest as well as statistical
231 analysis of the collected data. The software is only available for Windows.
232 The project initially open-source change to a closed-source model, but
233 the software is still under development.

234 1.3.2 Open-source software

235 Open-source software allows the user to read, modify, and distribute
236 the software. It is the preferred alternative to commercial software. From
237 a scientific perspective, using open-source software increase transparency
238 and lead to easier replicability of scientific results. From a development

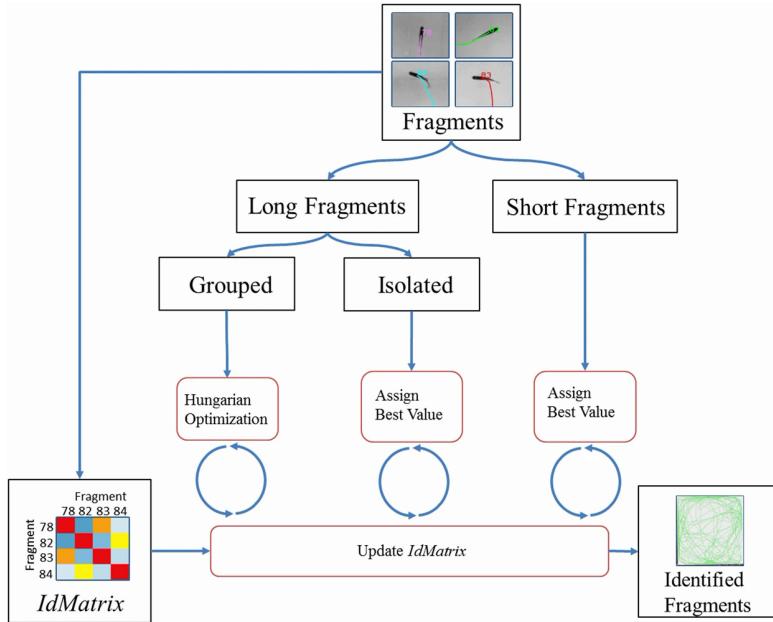


Figure 1.1: ToxId workflow chart.

standpoint, it leads to better code quality and fewer bugs. In general, no individual assistance service is provided. The collaborative development of most of these software allows the user to report bugs and participate in their development to help the community.

idTracker IdTracker [14] is a MATLAB library that allows to track multiple objects in video recording. It is based on the extraction of a "fingerprint" for each object thus a reliable association without errors propagation. The advantage of idTracker is that it can recognize an object over several videos and after a relatively long time, which can be useful to track individuals' behavior over several series of experiments.

IdTracker is solving amazingly well the propagation of errors problem during the association phase. However, it is limited by the number of objects it can track, currently about twenty, due to the movie's length necessary for extracting each object's "fingerprint". This task can go up to 30 minutes minimum for a high object density. The required image qual-

254 ity is an essential factor and must be at least 150 pixels per animal. The
255 computation time is relatively long, in the order of 0.5 to 2 seconds per
256 image, and requires a large RAM amount. The installation of idTracker can
257 be done without the need to buy MATLAB thanks to the Matlab Run Time
258 Compiler but only under Windows. Therefore, it is necessary to purchase
259 a MATLAB license for other platforms and have minimal knowledge of the
260 language to set up idTracker.

261 **DeepLabCut** DeepLabCut [16] is a framework that solves the so-called
262 "pose estimation" problem, which consists of finding an object and its
263 position, or part of an object, in an image. It can be directly related to
264 the SOT problem if the objects to be tracked are different, for example,
265 a right mouse ear and a mouse nose, which can then be found on each
266 image and then associated in the case where there is only one mouse.
267 In the case of several similar objects to be found and associated from
268 one image to another (MOT), this detection will have to be combined with
269 an association step to obtain the tracking. Even if DeepLabCut answers a
270 slightly different problem, it can, by its design, be coupled with an exter-
271 nal association algorithm to make a tracking software.

272 DeepLabCut is directly based on the feature detection algorithm of the
273 DeeperCut framework [20], specialized in the detection of human body
274 movements. The authors of DeepLabCut have studied this algorithm's per-
275 formance applied to the field of behavioral neuroscience, such as the de-
276 tection of mice snouts or drosophila legs. They have added tools to train
277 the algorithm easily and test its robustness.

278 It takes advantage of deep learning to solve the so-called estimation
279 pose problem. As a reminder, deep learning is a machine learning algo-
280 rithm that consists of training a neural network containing several layers.
281 In DeepLabCut, the network consists of several residual neural networks
282 (ResNets) pre-trained on the ImageNet database. The network is then fine-
283 tuned by training on images where the parts to be detected are annotated.
284 In the end, the algorithm gives the probability of presence of the object

285 in the image. The authors have shown that the performance is at least
286 as good as human detection and can be obtained with very little training
287 data (200 annotated images).

288 DeepLabCut, as previously mentioned, is a framework, and despite an
289 excellent documentation [21], it can be challenging to use it for a user
290 with little computer skills. The installation process lasts from 10 to 60
291 minutes and requires a GPU installation to get the most out of the soft-
292 ware. Besides, the algorithm requires a lot of computing power. To give an
293 idea, images of 682x540 pixels, analyzed with a last-generation GPU, lead
294 to an analysis speed of 30 frames per second. Without GPU, this time can
295 be multiplied by a factor of 10 or 100 [22].

296 We see that DeepLabCut is of great interest to find objects in an image
297 with precision. It is particularly aimed at behavioral neuroscience, allowing
298 complex movement tracking (e.g., hand fingers in a mouse). It will not
299 be suitable for users with little computer knowledge interested in more
300 extensive problems and with little data to process.

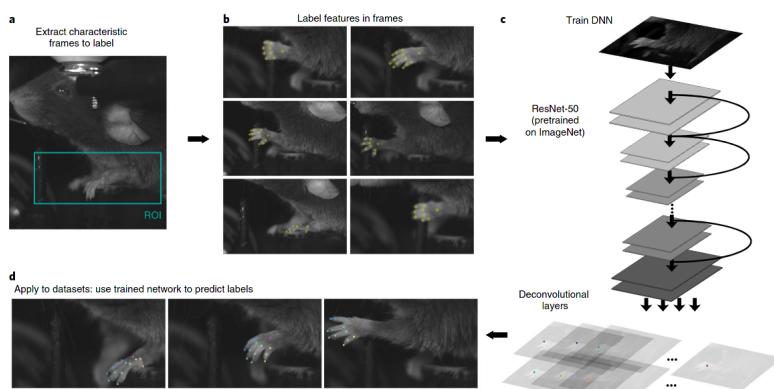


Figure 1.2: DeepLabCut workflow chart.

301 **idTracker.ai** IdTracker.ai [17] is a framework that allows tracking animals
302 with almost perfect accuracy. IdTracker.ai takes advantage of deep learn-
303 ing to carry out the association. In the first step, each object is segmented
304 by applying a threshold. A convolutional network classifies each detected

305 blob as containing a single object or several objects. Another convolutional
306 network finds the identity of each individual throughout the movie.

307 This system requires enough data to train the network that will rec-
308 ognize each individual. The authors found that robust tracking can be
309 obtained with only thirty isolated images of each individual. Therefore, it
310 is necessary to plan for a minimum of five hundred images for a dozen
311 individuals with a minimum of twenty-five frames per second. A resolu-
312 tion of three hundred pixels per animal is recommended for good track-
313 ing accuracy. A limiting factor of idTracker.ai is that it requires a lot of
314 computing time and a lot of RAM. The authors report about twenty min-
315 utes for processing a video with eight zebrafish and about six hours for
316 a hundred zebrafish on about two thousand high definition images. Even
317 if a UI is available to help the user, primary computer and programming
318 knowledge is required, and suitable hardware. The use of a GPU is strongly
319 recommended.

320 This software is suitable for users who want perfect and fully auto-
321 mated tracking from high-quality videos having a powerful computer. A
322 tool is integrated to review and correct the tracking, but the lack of ergon-
323 omy makes it sometimes difficult to use.

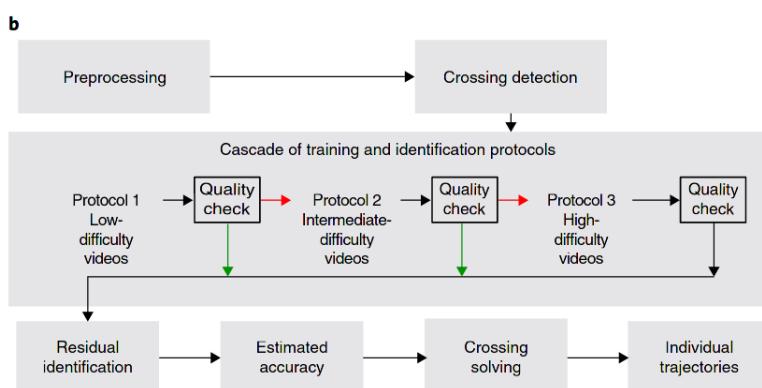


Figure 1.3: IdTracker.ai workflow chart.

324 1.4 FastTrack: an original approach

325 We have previously listed the most used tracking software in different
326 fields of science. We can see that a fast software requiring little computing
327 power, versatile (i.e., that can be applied to any systems with a variable
328 number of objects), easy to install, and open-source is missing. To fill this
329 void, we designed a software called FastTrack. This software contains two
330 distinct parts:

- 331 • An interface where standard image analysis procedures are imple-
332 mented to detect objects, and a tracking algorithm that allows keep-
333 ing the identity of objects from one image to another, fast and with
334 a low error rate.
- 335 • An ergonomic interface where the tracking can be checked and and
336 manually corrected if necessary.

337 We will notice here the difference in approach between FastTrack and ex-
338 isting software. Instead of developing a system that requires high comput-
339 ing power, which is slow but provides fully automated and highly reliable
340 results, FastTrack implements a fast and easy to set up, very generalized
341 method. The correction of the remaining errors is left to the user but can
342 be done natively in the software, the ergonomic interface allowing a fast
343 and efficient correction. This solution has several advantages, the first
344 one being that it does not require any programming knowledge. Any user
345 can perform a perfect analysis in a very short time. Moreover, it has been
346 shown that the post-processing work can be estimated by an analysis of
347 the geometrical and dynamic parameters of the studied system, which
348 allows the user to know if the software is adapted to his needs. For many
349 of the systems studied, the post-processing is only a quick check. If the
350 number of occlusions is too high, and a perfect tracking accuracy is nec-
351 essary without having to resort to manual correction, another solution
352 must be considered.

353 FastTrack is distributed under a free software license and implemented
354 in a modular and fully documented manner. Each user can thus modify the
355 software at his convenience or contribute to it. The tracking algorithm is
356 decoupled from the detection and correction interface, which makes it ex-
357 tremely easy to integrate FastTrack into an existing project. The software
358 is easily installed in less than 5 minutes and is compatible with Linux, ma-
359 cOS, and Windows. It can run on modest configurations and Single Board
360 Computer (SBC) such as the Raspberry Pi.

361 **Chapter 2**

362 **Design and implementation**

Testing can only prove the presence of bugs, not their absence.

Edsger W. Dijkstra

363 **2.1 Tools used**

364 The choice of tools and libraries used in designing software is paramount,
365 and several selection factors must be taken into account. The first criterion to consider is the license. We chose to put FastTrack under a free
366 license (GPL3), which implies that the language used and the libraries
367 must also be under compatible licenses. The choice of an open-source license is evident in the case of scientific software. The user can then check
368 how the software works, change it to adapt it to his needs, and share it.
369 The second criterion is to choose the libraries used carefully, considering
370 the future of the software so that the developers do not have to change
371 libraries if their capabilities prove insufficient as the software evolves.
372
373

374 Mature libraries offering long-term support are preferred.

375 In this perspective, FastTrack has been implemented in C++ using the
376 Qt [23] and OpenCV [24] libraries for the graphical interface and image
377 analysis, respectively. Unit tests are performed using the Google Test li-
378 brary.

379 C++ is a computer language created by Bjarne Stroustrup in 1985. Offer-
380 ing high performance, it is standardized by the International Organization
381 for Standardization (ISO). It is the language of choice for image analysis
382 applications and the creation of complex graphical user interfaces.

383 Qt is an open-source GUI library created by Haavard Nord and Eirik
384 Chambe-Eng, two physicists, in 1991 when developing ultrasonic image
385 analysis software. With extensive documentation and a large community,
386 it is very mature. It allows creating graphical user interfaces for Linux, Mac,
387 and Windows from the same source code.

388 OpenCV is an open-source image analysis library created by Intel in
389 1999. Very complete and efficient, it has become the reference in image
390 analysis for both academics and commercial applications.

391 Google test is a suite for automating unit tests in C++. OpenCV notably
392 uses it. The purpose of unit tests is to verify that each part of the program
393 works as expected. This practice has several advantages: detecting more
394 easily possible errors during the implementation of new features and fa-
395 cilitating software development when it grows in size to avoid any error
396 inclusions. This series of tests are automatically performed on each new
397 commit, see Section 2.3.1.

398 2.2 Implementation

399 FastTrack's operation can be divided into three parts: the detection of
400 objects, the association of objects from one image to another, and finally,
401 a correction step.

402 Each analysis begins with the opening of an image sequence or a video
403 file. The user can choose between two types of interfaces, an interactive
404 interface where he can only open one film at a time. It allows the user
405 to see, in real-time, the impact of parameters on the images, which facil-
406 itates the determination of the optimal analysis parameters. A second in-
407 terface allows many movies to be opened simultaneously, either by giving
408 a parameter file or selecting the parameters in the interface. It is useful
409 when the user wants to analyze many movies for which he already knows
410 the optimal analysis parameters.

411 Both interfaces can be used in a complementary way. The user can find
412 the optimal parameters with the interactive interface and then automate
413 the analysis of many movies by tracking them in batches in the software.

414 2.2.1 Detection

415 The purpose of the detection step is to extract each object's kinematic
416 parameters, which will be used later in the association step. FastTrack in-
417 cludes a collection of image analysis filters that allow the user to optimize
418 object detection without external software.

419 **Background Calculation** Each analysis starts by calculating a background
420 image. If the user already has a previously saved background image, he
421 can directly open it in the software. Otherwise, three calculation methods
422 are possible:

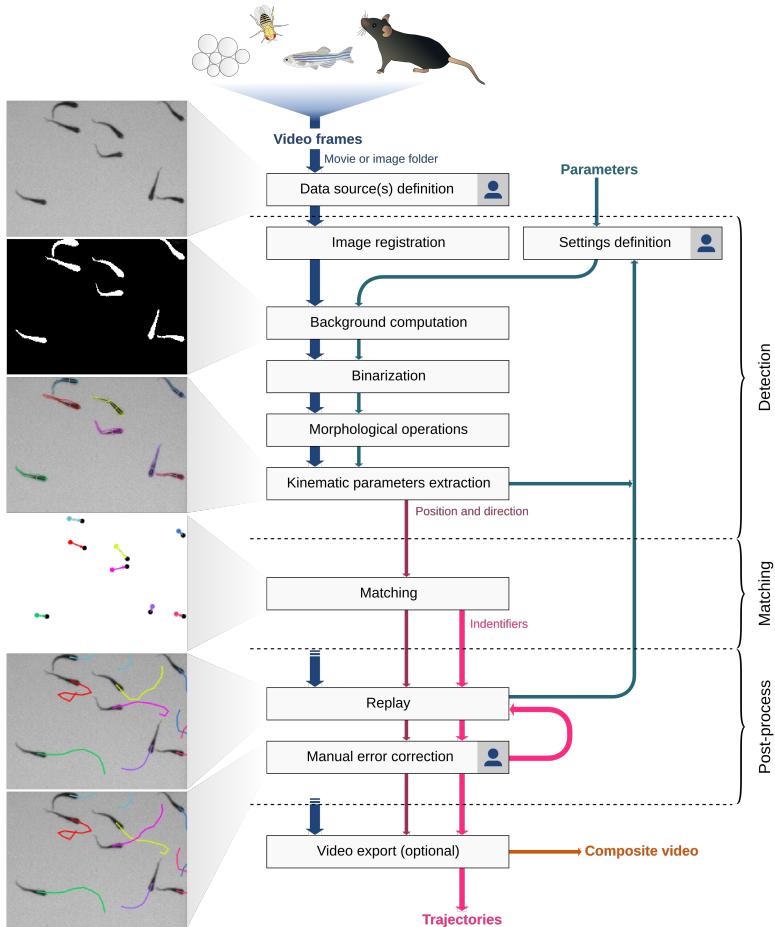


Figure 2.1: FastTrack workflow chart. The workflow divides into three mains parts: detection, matching, and post-processing. A indicates the few steps that require user input. Sample dataset: ZFJ_001.

- 423 • Projection of maximum intensity.
 - 424 • Projection of minimum intensity.
 - 425 • Projection of the average intensity.
- 426 All three methods are based on the same principle. The user chooses n images in the sequence. The software will make a projection of the stack along the time component, either the maximum, minimum or average of each pixel. In practice, the maximum (resp. minimum) will be projected

430 if the objects are darker (resp. lighter) than the background so that the
431 objects disappear and thus obtain the background. The user can make
432 the registration of each image before the projection in order to correct a
433 possible camera movement.

434 **Registration** The user can choose to register the images. Three meth-
435 ods are proposed in the software. Each method is implemented in a pyra-
436 midal way, i.e., the registration is first carried out on a degraded image
437 to roughly correct the displacement. The correction is then refined by
438 increasing the image quality until the original quality is reached. This
439 speeds up the process, as registration is often a relatively time-consuming
440 process.

441 The first method proposed is phase correlation. It allows correcting
442 the translational movements between two images using the Fourier the-
443 orem in the frequency domain. This method is swift but remains limited
444 to small translational movements only.

445 The second proposed method is the Enhanced Correlation Coefficient
446 (ECC) [25] method. In FastTrack, it is restricted to correcting translational
447 and rotational movements only. It consists of using the correlation coef-
448 ficient as a measure to find the best transformation between two images.
449 This method's advantage is that it is relatively fast since this non-linear
450 optimization problem can be solved linearly. It is efficient for noisy im-
451 ages and having photometric distortions.

452 The third method is a method based on the identification of key points.
453 It allows for correcting movements and deformations (homography). The
454 key points (about 500) are automatically determined on two images thanks
455 to the ORB algorithm [26]. These points are then associated two by two us-
456 ing the hamming distance. The RANSAC algorithm [27] is used to find the
457 best transformation between the two images. This method, more precise,
458 requires a sufficient image quality to be able to discern key points.

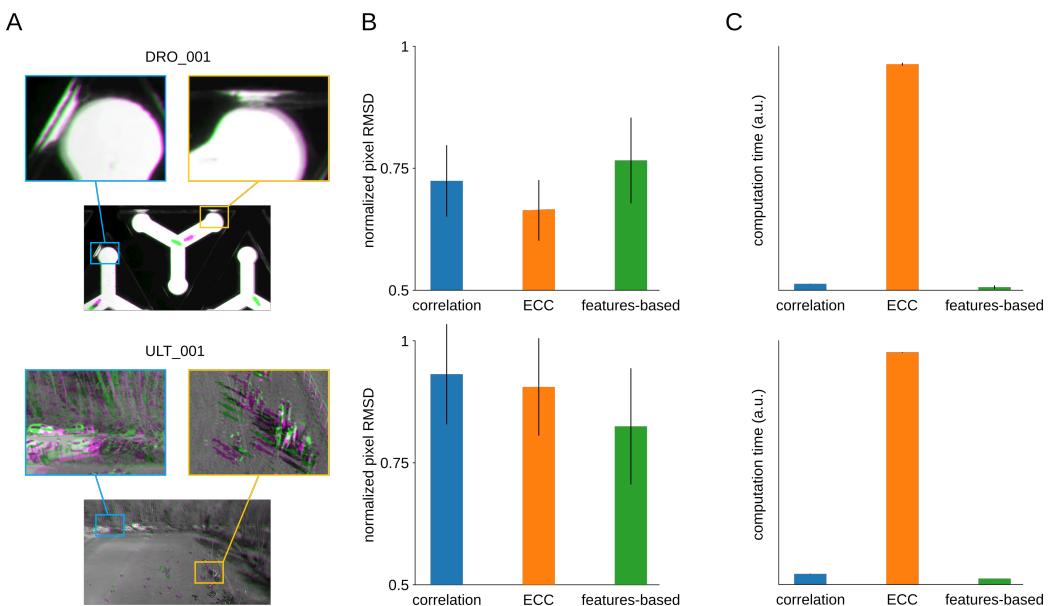


Figure 2.2: Image registration. Two recordings with severe drift are used for the benchmarking (top: *DRO_001*, bottom: *ULT_001*). **(A)** Comparison of a frame (magenta) with the first frame (green) and magnification of details in the scene. **(B)** The root mean square deviation (RMSD) of pixel intensities after registration onto the first image, averaged over all time frames and normalized by the RMSD without registration, for the three registration methods. Error bars: standard deviation across time frames. **(C)** The relative average computation time of the three registration methods, normalized by the total number of pixels in the movie (arbitrary units). Error bars: standard deviation across time frames.

459 Figure 2.3 provides a rough comparison of the performance of the
 460 three methods. Using two recordings of the dataset, we benchmarked
 461 both the accuracy – with the root mean squared difference (RMSD) of
 462 pixel intensities between the reference and the corrected image – and
 463 the relative computation time. Choosing the right method to obtain the
 464 best accuracy depends on each movie's characteristics. However, one can
 465 use the rule of thumb that if the objects to track occupy a large fraction
 466 of the total area, the best accuracy is more likely to be obtained by using
 467 ECC and using the features-based method otherwise. However, as shown
 468 in Figure 2.3-C, the ECC method is generally slower by an order of mag-
 469 nitude. Hence, we recommend using the features-based method in the

470 general case and long movies.

471 **Binarization** Each image is then binarized by subtracting the background
472 image and defining a threshold value. In the interactive mode, the user
473 can see the impact of the parameters on the image, which makes it eas-
474 ier to adjust the binarization threshold. The software also detects if the
475 background is darker (resp. lighter) than the objects allowing to have at
476 the end of this operation a binary image where the pixels belonging to
477 the object are equal to 1, and the pixels belonging to the background are
478 equal to 0.

479 **Morphological operation** A set of morphological operations (dilation,
480 erosion, opening, etc.) can be performed on the binary image to improve
481 detection and eliminate possible artifacts. Different shapes and sizes of
482 kernels are available.

483 **ROI** The user can select a region of interest and exclude the rest of the
484 image from the analysis. This speeds up the analysis process and avoids
485 the detection of interfering objects. In interactive mode, this ROI can be
486 drawn directly on the image.

487 **Sorting** To exclude objects that are too small (e.g., noise) or too large (e.g., two objects overlapping each other), the user must select two characteristic sizes. The objects are colored either red or green in the interactive mode depending on whether their size belongs to the selected range.

491 **Extracting kinematic parameters** Based on the binary images, the software will detect the contour of each object. An essential step in any tracking procedure is the extraction of the parameters used in the association step. It is generally with the choice of these quantities that the tracking

495 algorithms can differ to be more specialized for a given type of object.
 496 In FastTrack, the parameters extracted are the center of mass, the orien-
 497 tation, the area, and the object's perimeter. These quantities are quickly
 498 calculated and general enough to adapt to a wide variety of objects.

499 To do this, FastTrack calculates the object's equivalent ellipse from
 500 the second-order moments of the binary image. This procedure is accel-
 501 erated by directly using the contour thanks to Green's formula [?]. The
 502 object's orientation is given by the ellipse's major axis and is defined in
 503 the interval $[0; \pi]$. The direction in the interval $[0; 2\pi]$ is determined by
 504 projecting each object's pixel on the major axis of the equivalent ellipse,
 505 and calculating the skewness of the distribution of distances of these
 506 projected points to the center of mass. The skewness sign is a robust in-
 507 dicator of the object's asymmetry along its principal axis. For deformable
 508 objects, the previously calculated direction may be different from the di-
 509 rection of motion. For example, in the case of zebrafish, it bends its body
 510 periodically to move. Only the head is directed at the motion. This is why
 511 the object is decomposed into two equivalent ellipses. The user can then
 512 choose which ellipse best represents the direction of the movement.

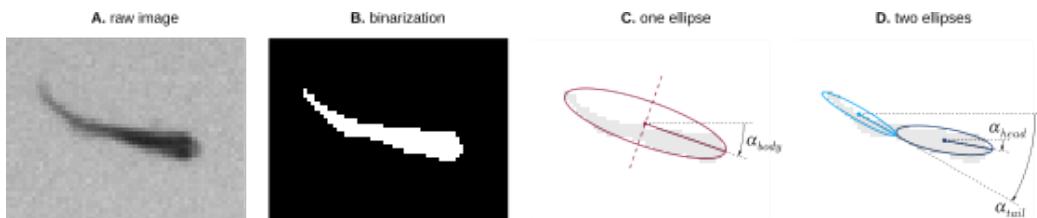


Figure 2.3: Detection Details of the detection phase for one object of the movie *ZFJ_001*. **(A)** raw image. **(B)** binarized image obtained by subtraction the back- ground image and applied a threshold. **(C)** equivalent ellipse of the object. **(D)** two equivalent ellipses, useful for a deformable object.

513 2.2.2 Association

514 The purpose of the association step is to keep the objects' identity
 515 from one image to another. To do so, FastTrack uses a method derived

516 from [13], which takes advantage of the fact that each object's position,
 517 area, perimeter, and direction changes very little from one image to an-
 518 other.

For each pair of objects (i, j) belonging to two successive images, two costs are calculated. The hard cost as follows:

$$\begin{cases} h_{i,j} = 1 & \text{if } r_{i,j} < h_d \\ h_{i,j} = \inf & \text{else} \end{cases}$$

519 with $r_{i,j}$ the distance between objects i and j, h_d a threshold represent-
 520 ing the maximum travel distance allowed between two successive im-
 521 ages. The hard cost allows discarding obvious impossible assignments
 522 to speed-up the computation. It is essential with a non-constant number
 523 of objects because it allows new objects entering the field of view to be
 524 assigned with new identities.

The soft cost is defined as follows:

$$c_{i,j} = \frac{r_{i,j}}{s_d} + \frac{\delta\alpha_{i,j}}{s_\alpha} + \frac{\delta a_{i,j}}{s_a} + \frac{\delta p_{i,j}}{s_p}$$

where $\delta\alpha_{i,j}$ is the angular difference, $\delta a_{i,j}$ the area difference and $\delta p_{i,j}$ perimeter difference between objects i and j. To compare these quantities expressed in different dimensions and magnitudes, we need to normalize them. We define the soft normalization coefficients: s_d , s_a , s_p and s_α . These coefficients represent the typical value of the parameter that they normalize. We can construct the cost matrix:

$$C_{i,j} = \begin{cases} c_{i,j} & \text{if } r_{i,j} < h_d \\ \inf & \text{else} \end{cases}$$

525 This cost matrix is, in general, rectangular because the number of objects
 526 can vary from one image to the following. A memory parameter can be
 527 selected to assign a new identity to an object that disappears on more
 528 than the selected number of images. In this case, we remove the row cor-
 529 responding to this object from the cost matrix. We want then to find the

530 best possible association. This problem is called the rectangular assign-
531 ment problem and can be solved exactly by using the Hungarian algo-
532 rithm see Annexe B. FastTrack uses the Kuhn-Munkres implementation in
533 C++ to solve it.

534 **2.2.3 Automatic tracking parameters**

535 Finding the optimal tracking parameters is necessary to have a track-
536 ing accuracy as good as possible. FastTrack can automatically determine
537 a neutral set of soft normalization factors s_r , s_α , s_A , and s_p to help the
538 user. These factors allow comparing terms of very different nature and
539 amplitude into a single cost function. The set of parameters automati-
540 cally found by FastTrack will give each term the same weight inside the
541 cost function. Therefore, the user must perform parameters' fine-tuning,
542 with some system insight, to get the best set of parameters possible.

543 It is intuitive to use the standard deviation of the increments of each
544 kinematic parameter. However, some trajectories are needed to estimate
545 the standard deviations. We set up an iterative, rapidly-converging algo-
546 rithm to perform this task.

547 Let us use *ZFJ_001*, a slightly oversampled movie, with many occlu-
548 sions and objects of different sizes to illustrate the algorithm's details.
549 For simplicity, let us use only the position, angle, and area as kinematic
550 parameters. There is no gain to expect by adding the perimeter param-
551 eter because objects' shapes are very similar. The Figure 3.1-A. shows a
552 snapshot of this movie.

553 To evaluate the distributions of dr , $d\alpha$, and dA , we start by tracking
554 the movie setting the hard parameters and random soft parameters. The
555 resulting distributions are shown in Figure 3.1-C to E. For kinematic pa-
556 rameters whose differential values can be positive or negative, the distri-
557 bution is fitted by a Gaussian function, and the soft parameter is set to

558 the standard deviation. For instance, with the angular difference $d\alpha$ the
 559 fit reads:

$$f(d\alpha) = \frac{1}{s_\alpha \sqrt{2\pi}} e^{-\frac{d\alpha^2}{2s_\alpha^2}} \quad (2.1)$$

560 and s_α (orange bar in Figure 3.1-D) is stored as the soft parameter to use
 561 during the next iteration. The computation of the soft parameter for the
 562 displacement s_r is different since distances can only be positive. Assuming
 563 that the displacements along the x and y axes follow two independent
 564 Gaussian processes, the resulting displacement follows a χ distribution
 565 with 2 degrees of freedom, and the fit reads (see Annexe C for the detailed
 566 derivation):

$$f(x) = \frac{x}{(\frac{s_r}{\sigma_0})^2} e^{-\frac{1}{2}(\frac{x}{\sigma_0})^2} \quad (2.2)$$

567 where s_r (orange bar in Figure 3.1-C) is stored as the soft parameter to use
 568 for the next iteration and a constant $\sigma_0^2 = 2 - \mu_0^2 = \frac{4-\pi}{2}$.

569 Once all soft tracking parameters have been derived from the distri-
 570 butions, the software recomputes new trajectories with these updated
 571 parameters. This iterative process, depicted in Figure 3.1-B, is run until
 572 the tracking parameters converge. In practice, the convergence is very
 573 fast, regardless of the initial position in the parameters space. We drew
 574 100 sets of seed parameters from uniform distributions spanning large in-
 575 tervals, and convergence has been attained in very few iterations for all
 576 parameters Figure 3.1-F.

577 FastTrack's implements this algorithm by taking the kinematic quanti-
 578 ties' sample standard deviation for a subset of 200 images in the movie
 579 to increase speed and efficiency. The convergence criterion implemented
 580 is that soft parameters should vary less than 10^{-3} .

581 To characterized the resulting tracking, we computed the number of

582 swaps with respect to the ground-truth:

$$P_{swap} = \frac{N_{swap}}{N_{obj} - n_{ap}} \quad (2.3)$$

583 with N_{swap} being the total number of swaps, N_{obj} the total number of ob-
584 jects on all frames and n_{ap} the number of times a new object appears.
585 If the number of objects is constant and noted n , then $n_{ap} = n$ and
586 $N_{obj} = nT$, with T the number of frames in the recording, such that P_{swap}
587 can be simplified:

$$P_{swap} = \frac{N_{swap}}{n(T - 1)} \quad (2.4)$$

588 P_{swap} converges very fast to a value that is nearly-optimal. For 77%
589 of the parameter sets P_{swap} is decreased or remain equal, with an aver-
590 age drop of 0.0119 (155% of the converged value), while for 23% of the
591 parameter sets P_{swap} is increased with an average rise of 0.0011 (14% of
592 the converged value). Thus, the expected difference is -0.0090 (116% of
593 the converged value) for this movie. Therefore, the automatic parameters
594 are an excellent starting point in the general case. The user can fine-tune
595 the weights given to the kinetic parameters to consider the specificities
596 of each movie.

597 We computed the converged soft parameters \hat{s}_r , \hat{s}_α and \hat{s}_A for several
598 sampling rates of $\tau > 1$ (Figure 3.1-H to J). We used these parameters to
599 track the ZFJ_001 movie at different τ and compute P_{swap} . A comparison
600 between P_{swap} and P_{inc} as a function of τ is shown in Figure 3.1-K. This
601 comparison illustrates that P_{swap} is a noisier measurement of a movie's
602 trackability than P_{inc} and confirms that the iterative algorithm produces
603 trajectories with a number of errors that is close to the statistical limit.

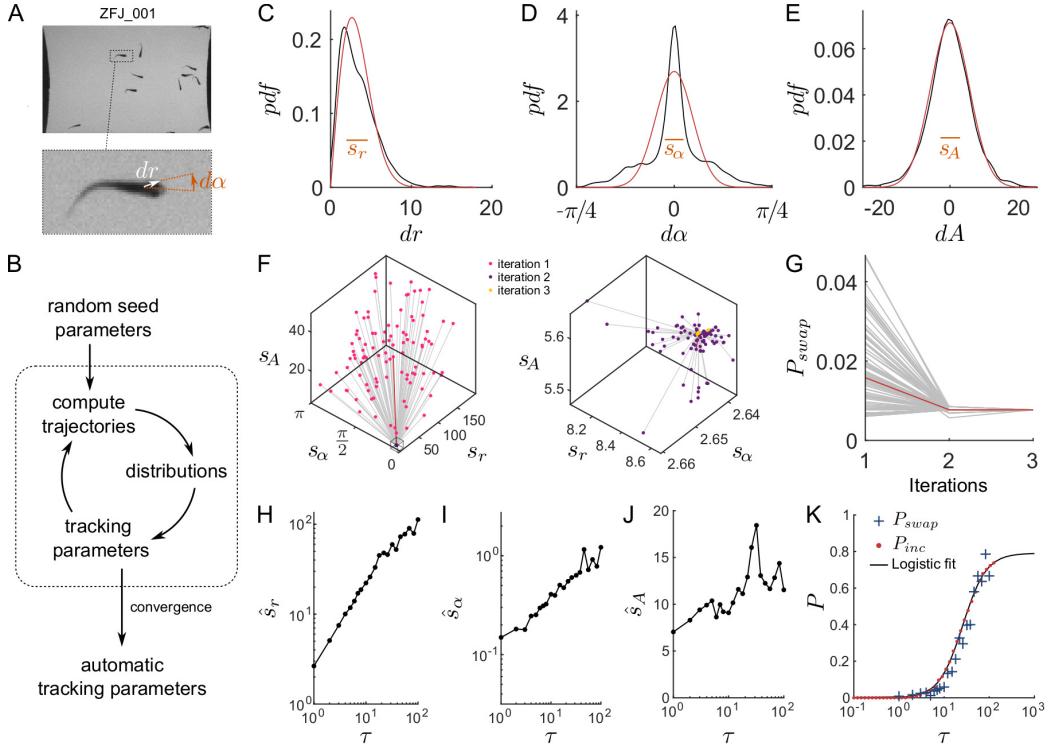


Figure 2.4: Automatic tracking parameters. **(A)** Snapshot and blow-up of *ZFJ_001* movie, with definition of \vec{dr} and $d\alpha$. **(B)** Scheme of the algorithm determining the tracking parameters automatically. **(C-E)** Distribution of displacements dr (in pixels), angular differences $d\alpha$ (in radians) and area differences dA (in pixels) when the default parameters of the software are used on *ZFJ_001*, for $\tau = 1$ (black). The corresponding χ and Gaussian fits are displayed in red. Orange bars: resulting soft parameters. **(F)** Evolution of s_r , s_α and s_A with algorithm iterations for *ZFJ_001*. Left: iterations 1 and 2; right: iterations 2 and 3. A hundred runs with random initial values are shown. The run with the software default parameters is highlighted in red. **(G)** Evolution of P_{swap} with algorithm iterations, same runs. **(H-J)** Evolution of the converged parameters \hat{s}_r , \hat{s}_α and \hat{s}_A as a function of the timescale τ for *ZFJ_001*. **(K)** Comparison between P_{swap} (blue crosses) obtained with the converged parameters and P_{inc} (red dots) for *ZFJ_001*. The solid black line is the logistic fit of P_{inc} .

604 2.2.4 Manual correction

605 FastTrack integrates a manual tracking correction tool. Once the anal-
606 ysis is completed, the result can be displayed in an ergonomic interface

607 created solely for this purpose. The user can replay the film by superim-
608 posing the results of the analysis on the original movie. The user can
609 interactively see each object's parameters. More importantly, the user
610 can also correct tracking errors by deleting objects or exchanging objects'
611 identity. This interface is designed with ergonomics and performance in
612 mind. Keyboard shortcuts and an on-the-fly selection of objects by click-
613 ing on the video allow the user to check and correct the analyses quickly.
614 It is also possible to record a film with the tracking results overlay su-
615 perimposed. This manual correction interface makes it possible to shift
616 the workload from the traditional pre-processing of data to the tracking
617 result's post-processing. In general, tracking software will reduce user in-
618 puts and improve tracking by using the raw images' pre-processing and
619 the conception of system-specific tracking algorithms. With FastTrack, the
620 pre-processing step is reduced to the minimum to remain general and be
621 applied to a wide variety of systems. To compensate for this, FastTrack
622 provides a correction tool. In the following chapter, we will see that this
623 method can save the user much time because the correction time is in
624 general lower than the conception and computational time of system-
625 specific tracking algorithms.

626 **2.2.5 Analysis**

627 A question that comes back a lot from users' feedbacks is what to do
628 after the tracking. FastTrack only takes care of the tracking, and statistical
629 data analysis is not implemented in the software. After the tracking, the
630 software generates several files containing the results and the tracking
631 parameters. The result file is named tracking.txt, and it contains the raw
632 data of the analysis with one image and one object per line. This format is
633 compatible with the most used analysis software (R, Python, Julia, spread-
634 sheet). Examples in Python and Julia are available in the documentation
635 to get started.

636 2.3 Deployment

637 2.3.1 CI/CD

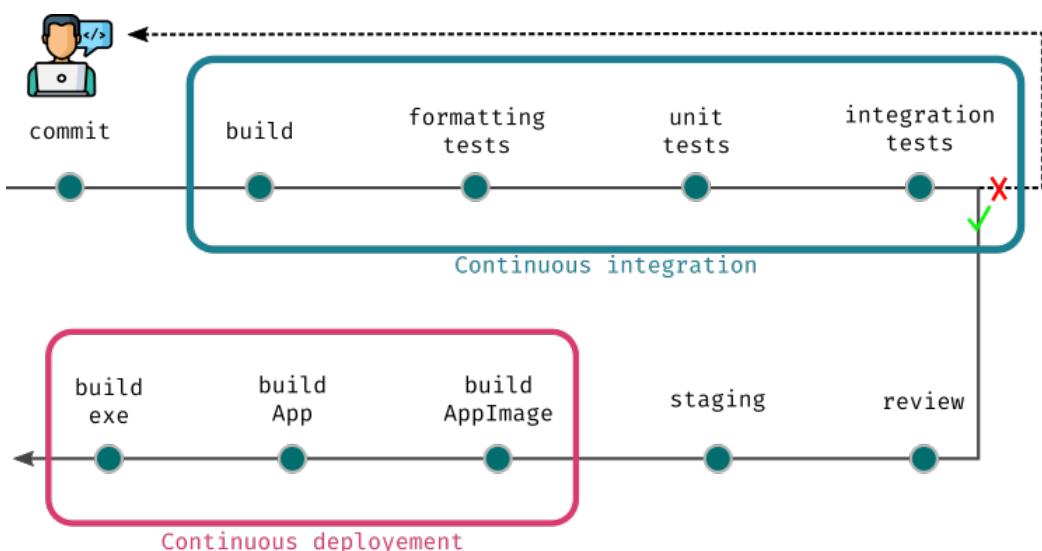


Figure 2.5: FastTrack CI/CD workflow. The CI/CD workflow is divided into two parts: the CI tasks (blue rectangle) and the CD tasks (red rectangle). The CI must be performed successfully in order to integrate the changes into the project.

638 The deployment is one part that should not be overlooked in soft-
639 ware design, and two aspects are crucial to consider. From the user's
640 point of view, the software must be easy to install on supported plat-
641 forms and with fewer bugs as possible. From the maintainer's perspective,
642 the deployment part must be easily achievable and reproducible so that
643 patches and new functionalities can be quickly integrated. From the devel-
644 oper's perspective, the source code's consistency and correctness have to
645 be tested at each change to avoid introducing bugs and facilitate collabo-
646 ration between developers. With this in mind, FastTrack follows the CI/CD
647 philosophy [28] [29] taking advantage of the new GitHub Actions system.

648 Continuous Integration (CI) is a set of practices designed to integrate
649 changes quickly into the project in an automated manner. It is coupled

650 with the automation of unit testing. FastTrack takes advantage of GitHub's
651 CI/CD system called Actions. With each new (commit¹) or new (pull-request
652 ²), a series of tests is automatically triggered. These tests will check the
653 proper functioning of the tracking algorithm and the formatting of the
654 source code. Only the changes that pass the tests can be integrated into
655 the project, which guarantees the reproducibility of the analyses and the
656 source code and documentation consistency.

657 Continuous Delivery (CD) automates the delivery of the software in
658 its final form. It allows changes to be quickly integrated into the software
659 without manually doing it for each supported platform. In the case of Fast-
660 Track, the CD is implemented using GitHub Actions, and a new version of
661 the software is compiled for Linux, macOS, and Windows with each new
662 commit that is integrated into the main branch. Stable versions of the
663 software are compiled at regular intervals of the development. This sys-
664 tem is a significant time-saver for multi-platforms software like FastTrack.
665 It allows the user always to have the latest patches and features available.
666 The developers can collaborate easily on the project, and the maintainer
667 can quickly produce binaries for the supported platforms.

668 FastTrack natively supports the three most commonly used platforms:
669 Linux systems with an AppImage that supports all distributions, Windows
670 with an installer, MacOS with an App. The latest stable version can be
671 downloaded from the website <http://www.fasttrack.sh>, the nightly build
672 version from <https://github.com/bgallois/FastTrack/releases>. The pro-
673 cedure to compile the software itself is available in the developer's doc-
674 umentation.

¹Action to send the list of changes made in the version management system

²Action to request the addition of changes to the project

675 2.3.2 Documentation

676 FastTrack offers extensive documentation that covers the installation
677 and utilization of the software. Developer documentation with a docu-
678 mented API and a compilation guide is also available for users wanting
679 to integrate FastTrack in their software or workflow.

680 **User** User documentation is available at <https://www.fasttrack.sh/UserManual/docs/intro.html>. This documentation is generated from the
681 project, and users can contribute to it at <https://github.com/bgallois/FastTrack/>.
682 It contains all the information needed to use the software and instruc-
683 tional videos to help the user get started with the software.
684

685 **Developer** Developer documentation is available at <https://www.fasttrack.sh/API/index.html>. It is automatically generated by the Doxygen soft-
686 ware from the documentation in the FastTrack source code. It contains
687 all the information necessary for developers who want to modify or con-
688 tribute to FastTrack.
689

⁶⁹⁰ Chapter 3

⁶⁹¹ Movies dataset

⁶⁹² To demonstrate that FastTrack can analyze movies from various sys-
⁶⁹³ tems, we have compiled a database of movies named TD^2 . This database
⁶⁹⁴ can be downloaded at <https://data.ljp.upmc.fr/datasets/TD2/>. The
⁶⁹⁵ films either come from data already published in the literature or pro-
⁶⁹⁶ vided to us by the authors themselves. All the movies are under a CC-
⁶⁹⁷ BY-NC-SA license. Each movie is identified by a 3-letter code defining the
⁶⁹⁸ system (e.g., ACT: active matter, ZFA: zebrafish adult...) and three digits
⁶⁹⁹ to index films from an identical system. TD^2 currently regroups 41 films,
⁷⁰⁰ including different types of objects of very different nature and size

- ⁷⁰¹ • 7 species of animals from fish to flies,
- ⁷⁰² • cells,
- ⁷⁰³ • active particles,
- ⁷⁰⁴ • microfluidic drops,
- ⁷⁰⁵ • macroscopic objects such as ultimate players or cars.

⁷⁰⁶ A video giving a quick overview of all the systems used is available at <http://www.fasttrack.sh/images/illustrations/mockups/trackingExample.webm>.

708 Another essential aspect to consider is the number of objects per film
 709 and their possible appearances, disappearances, and overlaps. In 22 films
 710 out of 41, the number of objects is variable, and objects come and go
 711 out of the camera field during recording. In 19 films out of 41, objects
 712 may overlap, creating an occlusion phenomenon that the software has to
 713 manage to preserve the identity of the objects.

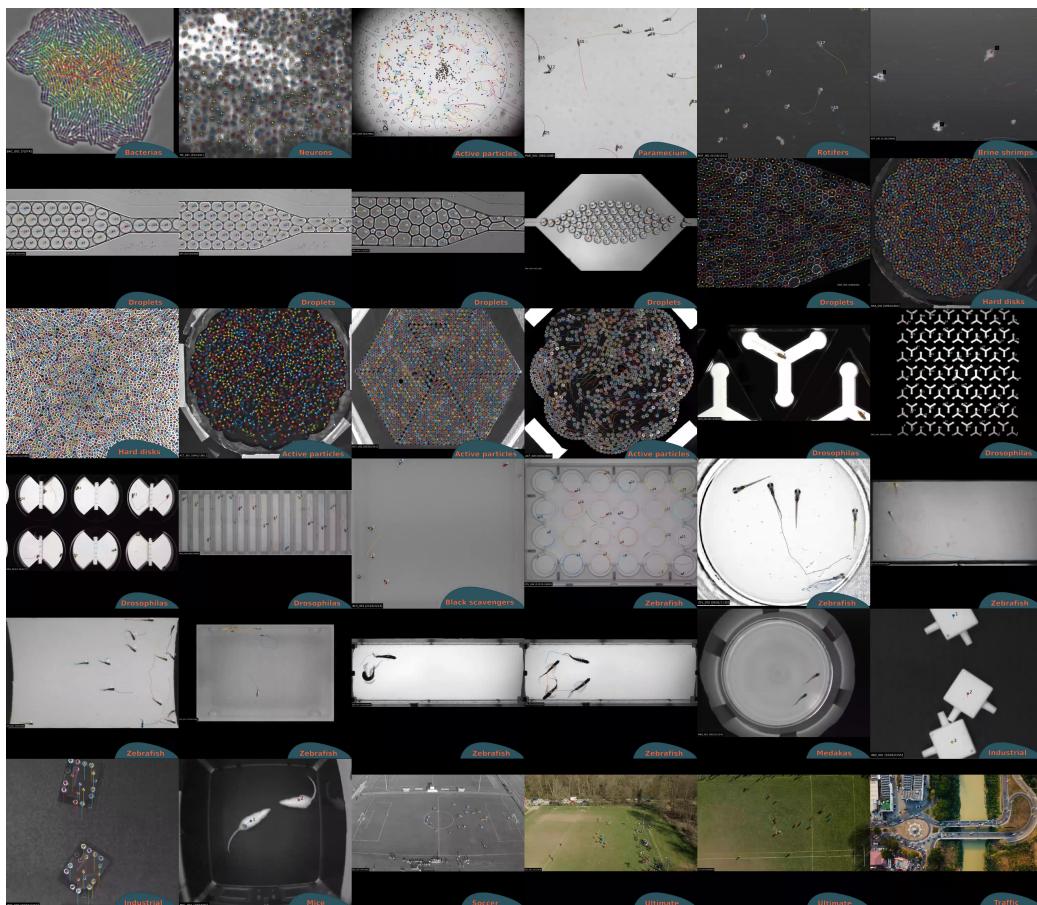


Figure 3.1: TD² Thumbnail of the TD² dataset.

S1 Table. Two-Dimensional Tracking Dataset.

Description and credentials of the data that have been used for testing the FastTrack software. All movies in the dataset can be downloaded at <http://data.ljp.upmc.fr/datasets/TD2>.

Identifier	Color	Number of images	Image size	Number of objects	Object type	Short description	Authors	Reference	Overlaps	Preprocess
ACT_001	No	1,001	2040x2040	1141	Active particles	Vibrated polar monodisperse discs in a liquid-like configuration.	Olivier Dauchot	17	No	No
ACT_002	No	1,001	1121x996	1085	Active particles	Vibrated polar monodisperse discs in a crystalline-like configuration.	Olivier Dauchot	18	No	Yes
ACT_003	No	1,000	1728x1728	853	Active liquid	Vibrated polar monodisperse discs.	Olivier Dauchot	19	No	Yes
ACT_004	No	998	4096x3072	Variable	Active droplets	Motion of water droplets in an oil-surfactant medium of squalane and monoolein.	Olivier Dauchot	20	No	No
ART_001	No	2,000	1280x1024	Variable	Brine schrimps	Brine shrimps (<i>Artemia Salina</i> nauplii) swimming.	Raphaël Candelier	15	Yes	No
BAC_001	No	74	907x866	Variable	<i>Bacillus Subtilis</i>	Growing colony of <i>Bacillus Subtilis</i> .	Lydia Robert	<i>Unpublished</i>	Yes	Yes
BLS_001	No	2,213	1024x1024	7	Black Scavengers	Black scavengers walking in an arena.	Juan Pablo Busso	<i>Unpublished</i>	No	No
DRO_001	Yes	1,803	640x360	3	Adult fruit flies	Blow-up of three fruit flies (<i>D. Melanogaster</i>) in an Y-maze assay.	Benjamin de Bivort	<i>Unpublished</i>	No	No
DRO_002	No	1,589	640x480	91	Adult fruit flies	Fruit flies (<i>D. Melanogaster</i>) walking in an Y-maze assay. Very low resolution.	Benjamin de Bivort	21	No	Yes
DRO_003	Yes	927	1280x720	12	Adult fruit flies	Fruit flies (<i>D. Melanogaster</i>) in a multi-chamber assay.	Benjamin de Bivort	<i>Unpublished</i>	No	No
DRO_004	No	3,490	602x228	15	Adult fruit flies	Fruit flies (<i>D. Melanogaster</i>) in a multi-tubes odor assay.	Benjamin de Bivort	22	No	No
DRP_001	No	700	1224x476	Variable	Droplets	Round droplets evolving in a narrowing microfluidic channel.	Lea-lætitia Pontani	16	No	No
DRP_002	No	600	1224x477	Variable	Droplets	Polyhedral droplets evolving in a narrowing microfluidic channel.	Lea-lætitia Pontani	<i>Unpublished</i>	No	No
DRP_003	No	835	1223x434	Variable	Droplets	Polyhedral droplets evolving in a narrowing microfluidic channel.	Lea-lætitia Pontani	<i>Unpublished</i>	Yes	No
DRP_004	No	299	640x360	Variable	Droplets	Six-layers pattern of droplets in a diverging/converging microfluidic channel.	Bibin M. Jose, Thomas Cubaud	23	No	Yes
DRP_005	No	660	960x730	Variable	Droplets	Edge-fluorescent droplets evolving in a narrowing microfluidic channel.	Lea-lætitia Pontani	<i>Unpublished</i>	No	Yes
GRA_001	No	1,001	2040x2040	2043	Hard disks	Vibrated isotropic bidisperse discs.	Olivier Dauchot	<i>Unpublished</i>	No	Yes
GRA_002	No	1,009	900x900	Variable	Hard disks	Motion of dense bidisperse hard particles close to jamming.	Raphaël Candelier, Olivier Dauchot	24	No	Yes
GRA_003	No	165	1470x1469	Variable	Vibrated grains	Motion of a pulled intruder in a dense set of bidisperse particles.	Raphaël Candelier, Olivier Dauchot	24	No	Yes
HXB_001	No	1,001	2040x2048	10	Hexbugs	Tiny robots vibrated in a parabolic arena.	Olivier Dauchot	25	Yes	N/A
IND_001	No	2,255	512x512	Variable	Plastic parts	Plastic connectors for microfluidics moving on a conveyor belt. Some are isolated, some are overlapping.	Raphaël Candelier	<i>Unpublished</i>	Yes	No
IND_002	No	2,463	512x512	Variable	PCB	Small printed circuit boards (PCB) moving on a conveyor belt. Some are isolated, some are overlapping.	Raphaël Candelier	<i>Unpublished</i>	Yes	Yes
MED_001	No	2,134	640x480	2	Adult Medakas	Two fish (<i>Oryzias latipes</i>) following a moving visual pattern.	Hideaki Takeuchi	26	Yes	No
MIC_001	No	693	854x480	2	Adult mice	Two white mice interacting in a dark arena.	Noldus Information Technology	<i>Unpublished</i>	Yes	No
PAR_001	No	1,000	1024x1024	Variable	Paramecia	Paramecia swimming in an open field.	Alexis prevost	<i>Unpublished</i>	Yes	No
ROT_001	No	1,311	800x800	Variable	Rotifers	Rotifers (<i>Rotifera</i>) swimming in an open field.	Raphaël Candelier	15	Yes	No
SOC_001	No	600	1920x1080	23	Humans	Aerial view of a soccer game.	Raphaël Candelier	<i>Unpublished</i>	Yes	Yes
TIS_001	No	241	301x301	Variable	Neurons	Developing neurons in the hindbrain of a GCaMP3 3dpf zebrafish larva. Total acquisition time: 30 min.	Raphaël Candelier	<i>Unpublished</i>	No	Yes
TIS_002	No	241	301x300	Variable	Neurons	Developing neurons in the optic tectum of a GCaMP3 3dpf zebrafish larva. Total acquisition time: 30 min.	Raphaël Candelier	<i>Unpublished</i>	No	Yes
TRA_001	Yes	1,171	1280x720	Variable	Vehicles	Aerial view of traffic.	No Copyright Footage	<i>Unpublished</i>	No	Yes
ULT_001	Yes	383	1280x720	Variable	Humans	Aerial oblique view of a sequence of Ultimate (flying disk) game.	Paulin Huger	<i>Unpublished</i>	Yes	Yes
ULT_002	Yes	953	1280x720	Variable	Humans and disk	Aerial top view of a half-field Ultimate (flying disk) game.	Paulin Huger	<i>Unpublished</i>	Yes	Yes
ZFA_001	No	15,000	1920x1080	5	Adult zebrafish	Five adult zebrafish interacting.	Gonzalo G. de Polavieja	13	Yes	No
ZFA_002	No	2,000	1218x482	2	Adult zebrafish	Two long-fin (TL) and normal fin (AB) zebrafish interacting.	Benjamin Gallois	<i>Unpublished</i>	Yes	No
ZFA_003	No	2,000	1217x471	3	Adult zebrafish	Three Tupfel long-fin (TL) zebrafish interacting.	Benjamin Gallois	<i>Unpublished</i>	Yes	No
ZFA_004	No	2,000	1212x472	4	Adult zebrafish	Four zebrafish (TL, AB) interacting.	Benjamin Gallois	<i>Unpublished</i>	Yes	No
ZFJ_001	No	200	524x338	14	Juvenile zebrafish	Three weeks-old zebrafish swimming in a flow.	Benjamin Gallois	<i>Unpublished</i>	Yes	No
ZFL_001	No	72,000	736x736	39	Larval zebrafish	Two hours of 6 d.p.f. larvae freely swimming in an enclosed arena, recorded at 10Hz.	Raphaël Candelier	<i>Unpublished</i>	Yes	N/A
ZFL_002	No	1,130	721x696	5	Larval zebrafish	High-speed imaging of the response of confined zebrafish larvae to an acoustic pulse. Framerate: 1kHz.	Raphaël Candelier	<i>Unpublished</i>	Yes	No
ZFL_003	No	1,870	1000x500	3	Larval zebrafish	Three larval zebrafish attracted by a flow with a dye.	Raphaël Candelier	<i>Unpublished</i>	Yes	No
ZFL_004	No	14,830	1024x768	24	Larval zebrafish	Larval zebrafish in a 24-multiwell plate.	Elim Hong, Margherita Zaupa	<i>Unpublished</i>	No	Yes

Chapter 4

Results

4.1 Performance

To assess FastTrack's performance, we ran a benchmark comparing FastTrack, Idtracker.ai, and ToxTrac. These software have substantial intrinsic limitations compared to FastTrack. Both require an acceptable framerate and image quality, with sufficient contrast and number of pixels per object and a constant number of objects in the movie that must be defined before the tracking. The benchmark was performed on a dataset consisting of a selection of videos provided with each software, and some movies from the *TD*² dataset that meet the three software requirements. *idtrackerai* video example and 100Zebra are available on the idtracker.ai website <https://idtrackerai.readthedocs.io/en/latest/data.html>. Guppy2, Waterlouse5, and Wingedant on the ToxTrac SourceForge <https://sourceforge.net/projects/toxtrac/files/ScientificReports/>. Movies provided in image sequence format were converted losslessly to a video format using FFmpeg since idtracker.ai and ToxTrac could not directly process image sequences. *DRO_002* and *ACT_002* were preprocessed with a custom script to detect the objects before performing the tracking. Also, only the first 100 images of *DRO_002* were used to reduce the computing time.

735 The benchmark between idtracker.ai and FastTrack was performed on
736 a workstation with an Intel i7-6900K (16 cores), 4.0 GHz CPU, an NVIDIA
737 GeForce GTX 1060 with 6GB of RAM GPU, 32GB of RAM, and an NVMe SSD
738 of 250GB running Arch Linux. The parameters were set by trials and er-
739 rors inside the graphical user interface of the two software. The tracking
740 duration was recorded using a script calling the software command-line
741 interface. The average tracking duration and the standard deviation were
742 computed over five runs except for *DRO_002* (2 runs) and *ACT_002* (1
743 run) due to the very long processing time. Idtracker.ai was evaluated with
744 and without GPU capability except for *100Zebra*, *DRO_002*, and *ACT_002*
745 due to the very long processing time.

746 The benchmark between ToxTrac and FastTrack was performed on a
747 computer with an Intel i7-8565U (4 Cores), 1.99 GHz CPU, 16 GB of RAM,
748 and an NVMe SSD of 1 TB running Windows 10. The parameters were set
749 by trials and errors in the graphical user interface. The average tracking
750 duration and the standard deviation were computed over five runs using
751 each software's built-in timer feature. The accuracy was evaluated man-
752 ally using the built-in review feature implemented in each software. The
753 number of swaps and the number of non-detected objects were counted
754 in each movie, and occlusion events were ignored in this counting.

The accuracy was computed as follows:

$$A = \frac{n_{obj}n_{img} - (2N_{swap} + N_{undetected})}{n_{obj}n_{img}}$$

755 with N_{swap} the number of swaps, $N_{undetected}$ the number of non-detected
756 objects, n_{obj} the number of objects, and n_{img} the number of images. For
757 *100Zebra*, the accuracy was computed only over the 200 first images. All
758 the results are presented in Figure ???. As expected, FastTrack is several
759 orders of magnitude faster than idtracker.ai and significantly faster than
760 ToxTrac on all tested videos. That is mainly due to the method used, id-
761 tracker.ai using deep learning and ToxTrac cost optimization and the iden-
762 tity preservation algorithm. All software performed exceptionally well in
763 terms of accuracy, except idtracker.ai on *ZFJ_001* probably because the

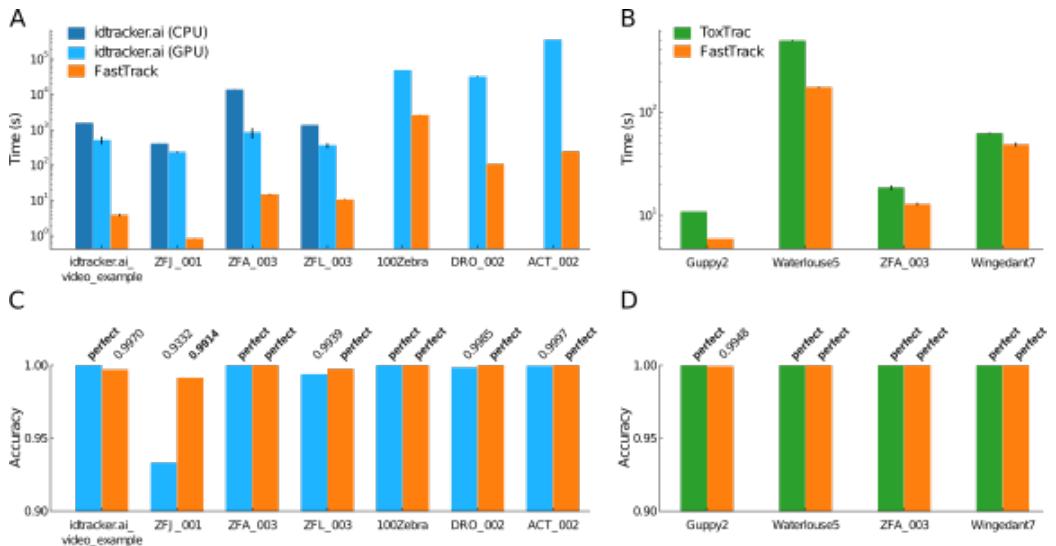


Figure 4.1: Benchmark of FastTrack, idtracker.ai, and ToxTrac. **(A-B)** Comparison of the computation time for the tracking of various movies with the same workstation. Whenever possible, CPU and GPU variants of idtracker.ai have been run. Only the first 100 images of *DRO_002* have been used. **(C-D)** Accuracies of the resulting trackings. "perfect" means an accuracy of exactly 1. The trajectories computed by the CPU and GPU variants of idtracker.ai being rigorously similar, we only show the GPU results. For *100Zebra*, the accuracy was computed by taking into account only the first 200 images.

764 resolution is not good enough. FastTrack's ergonomic post-processing interface can be used to reach a perfect tracking accuracy within a few more minutes. This built-in manual correction is not possible in ToxTrac and lacking ergonomy in Idtracker.ai.

768 Altogether, FastTrack offers many assets compared to idtracker.ai and 769 ToxTrac. The software is more versatile than its concurrents and more 770 comfortable to use. The total time spent to track a movie is globally lower, 771 in some cases by orders of magnitude, without sacrificing tracking accuracy. 772

773 4.2 Dataset classification

774 Analyzing movies from systems as different as those compiled in TD^2
775 is a real challenge. That is partly due to the recording conditions that can
776 be very diverse and make object detection more complex. Two recurring
777 difficulties can be discerned: variations in illumination (e.g., reflection in
778 GRA_001, shadows in SOC_001) and overlapping objects (e.g., HXB_001).

779 In movies from the academic world, systems are often designed to
780 limit or circumvent these two difficulties. It is common to find movies
781 with a uniform and constant illumination. Also, quasi-2D confinement and
782 a restricted number of objects in the camera field reduce the number of
783 occlusions.

784 In TD^2 , 23 movies have an illumination good enough to be analyzed
785 directly with FastTrack. The others had to undergo a specific individual
786 pre-processing before being analyzed. Two movies with too many occlu-
787 sions were discarded (HXB_001 and ZFL_001) because they could not be
788 analyzed with FastTrack. The remaining 39 films could be analyzed with
789 FastTrack without difficulty. The Kuhn-Munkres algorithm being of com-
790 plexity $O(n^3)$ the calculation time is generally quite fast. Each film was
791 then manually corrected using the built-in tool to get the ground-truth
792 tracking.

793 FastTrack is designed to keep the post-processing phase as light as
794 possible. However, this phase workload varies greatly depending on the
795 movie being analyzed. This workload can be quickly estimated for a given
796 film by computing what we call the probability of incursion.

797 First, we define the incursion as the exit of an object from its Voronoi
798 cell (see Annexe A), defined at a time t , after a travel time τ . The number
799 of incursions depends on

- 800 • the distribution of the displacements,

- 801 • the density of objects,
- 802 • the geometry of the Voronoï cell
- 803 • the degree of motion alignment of the displacements.

804 To consider the objects' density, we defined the reduced length $\rho = r\sqrt{d}$
 805 where r is the length and d the density. We remark that typically $\rho = 1$
 806 is corresponding to the length between two objects, and $\rho = 0.5$ is the
 807 length between an object and its Voronoï cell boundary.

Assuming that the dynamic is uncorrelated with the geometric properties of the Voronoï cells, we can write the incursion probability as follows:

$$P_{inc} = \int_0^{\inf} R(\rho)p_{inc}(\rho) d\rho$$

808 where $R(\rho)$ the distribution of the reduced displacement at the timescale
 809 ρ , and $p_{inc}(\rho)$ the geometrical probability of incursion.

810 $p_{inc}(\rho)$ depends only on the geometrical properties of the objects' arrangement.
 811 We can calculate p_{inc} by taking a Voronoï cell and determining
 812 the proportion of the angles for which a displacement of ρ implies an incursion
 813 in a neighboring Voronoi cell. In other words, see Figure 4.2, we
 814 take a circle of radius ρ centered on the object and count Σ the proportion
 815 of the circle that lies outside the Voronoï cell. That will give us $p(\rho) = \frac{\Sigma(\rho)}{2\pi}$
 816 the geometric probability of incursion for one cell. Then, to take into account
 817 the diversity of size and shape of Voronoï cells, we average over all
 818 the cells of the movies $p_{inc}(\rho) = \langle p(\rho) \rangle_{cells}$.

819 Intuitively, we see that p_{inc} goes from 0 when $\rho - > \inf$, to 1 when
 820 $\rho >> 1$. The precise shape of the geometric probability is sensitive to the
 821 density of objects, compact (e.g., ACT_002), sparse (e.g., PAR_001), and to
 822 the overall size of the system when walls restrict it (e.g., ZFA_001).

823 The distribution $R(\rho)$ is shown in Figure 4.3.B for three timescales τ .
 824 A graphical way of calculating P_{inc} is to take the intersection of the ar-

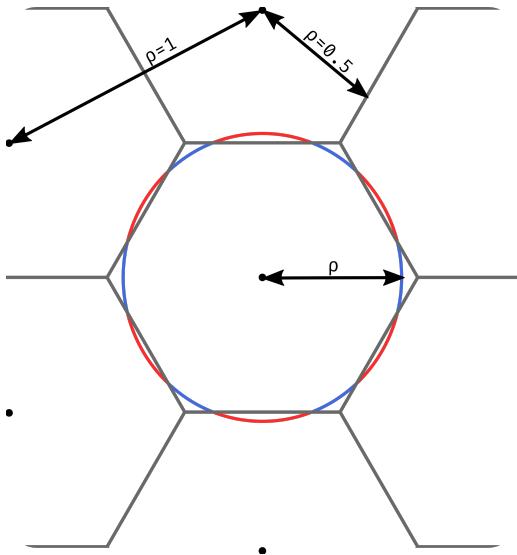


Figure 4.2: Geometric probability of incursion: Voronoï cells in black and objects black dots with typical ρ length. Circle of radius ρ with portions inside the Voronoï cell in blue and outside in red. The geometric probability for one cell is computed as $p(\rho) = \frac{\Sigma_{out}(\rho)}{\Sigma_{out}(\rho) + \Sigma_{in}(\rho)} = \frac{\Sigma_{out}(\rho)}{2\pi\rho}$

eas under $R(\rho)$ and $p_{inc}(\rho)$. In the regime where $R(\rho)$ and $p_{inc}(\rho)$ are well-separated, the resulting value of P_{inc} are low but highly sensitive to the number of swaps in the tracking. Indeed, the swaps create a bump in R at values of ρ close to one that can artificially increase P_{inc} of several orders of magnitude. Unless the ground-truth trajectories are accessible, the single value of P_{inc} at $\tau = 1$ can not be used as a measure for a movie's trackability.

A timescale-varying analysis will allow us to extract more robust quantifiers. As $p_{inc}(\rho)$ does not depend on τ and $R(\rho)$ is shifted to the high values of ρ when τ increases, we can expect that $P_{inc}(\tau)$ has a sigmoid-like shape. We thus computed P_{inc} for various τ . If $\tau > 1$ we take integer values (i.e. keep one frame every τ), and if $\tau < 1$ we linearly interpolated the displacements (i.e. multiplied ρ by τ). We represented the results in Figure 4.3.C for the 39 movies that could be tracked in the dataset.

Strikingly, all P_{inc} followed a logistic curve when τ is log-scaled. There-

fore we used a fit of the form:

$$P_{inc} = \frac{L}{1 + e^{-k(\log(\tau) - x_0)}}$$

and, noting $\tau_0 = e^{x_0}$, the fitting function can be rewritten as:

$$P_{inc} = \frac{L}{1 + \frac{\tau_0}{\tau} k}$$

The fits are shown in Figure 4.3, and are valid for all the movies in the dataset. We can make all fitting curves collapse on a single master curve. We show in Figure 4.3.D that $\frac{P_{inc}}{L}$ plotted as a function of $k \log(\frac{\tau}{\tau_0})$ follows the standard logistic sigmoid function

$$f(x) = \frac{1}{1 + e^{-x}}$$

839 .

An exciting outcome of this approach is determining the framerate at which experiments should be recorded. It is indeed a recurrent experimental question. A high temporal resolution is preferable to reduce the number of incursions and ease the tracking. However, it may not always be accessible (e.g., limited sensor rate, intense illumination required as the exposure time drops) and generates large amounts of images to store and process. A low temporal resolution can make the tracking difficult by increasing the number of incursions. We define τ_1 , the timescale at which P_{inc} reaches the inverse of the total number of objects on all frames N_{obj} , i.e., the probability of a single incursion in the whole movie. As τ_1 defines the onset of incursions and the possibility of swaps in the tracking procedure, it can be used to indicate each movie's sampling quality. Movies with $\tau_1 < 1$ already have incursions at the current framerate and are thus undersampled. Whereas for movies with $\tau_1 > 1$, the current framerate can be degraded without triggering incursions and are thus oversampled. Besides, τ_1 is directly the resampling factor that one should use to have minimal movie size without generating incursions. Using ??, it reads:

$$\tau_1 = \tau_0 (LN_{obj} - 1)^{\frac{1}{k}}$$

840 We computed and ordered the values of τ_1 in Figure 4.3.D for the whole
841 dataset. It appears that three quarters (30) of the movies are oversampled.
842 Any difficulty in the tracking should not be expected concerning incur-
843 sions. On the other hand, nine movies are undersampled. These record-
844 ings were already known to be difficult to track, three of them (*ACT_003*,
845 *ACT_004*, and *GRA_003*) have required specific algorithms for analysis,
846 and two (*BAC_001*, *ZFA_001*) required dedicated software.

847 Then, we tested to what extent this characterization is robust to swaps
848 in the trajectories. Starting from the ground truth trajectories of *ACT_002*,
849 we degraded the tracking quality by introducing random swaps between
850 neighboring objects. This process is controlled by a degradation rate δ ,
851 defines as the number of artificial swaps divided by the total number of
852 objects on all frames. Such a degradation affects the small timescales
853 more severely, and the multi-scale approach takes on its full interest. As
854 depicted in 4.3.F, the fits of $P_{inc}(\tau)$ are insensitive to degradation up to a
855 remarkably high-level of $\delta \approx 10^{-3}$. Therefore, even poor-quality tracking
856 can be used as an input for this method. As long as the distribution of dis-
857 placements is only marginally affected, the output remains unchanged.

858 4.3 Parameters optimization

859 One may also want to determine the optimal tracking parameters, i.e.,
860 with a P_{swap} close to 0 as possible. Provided that the ground-truth is
861 known for at least one movie of a system, for example, by a careful manual
862 post-processing. It is possible to leverage FastTrack's speed to explore the
863 parameters space and minimize P_{swap} . The optimized parameters found
864 that way can be used to track other similar movies with a minimal error
865 rate. The workflow of the method is depicted in the Figure 4.4-A. As the
866 exploration of the whole parameters space requires to perform at least
867 thousands of trackings, such an approach is only made possible by the
868 command-line interface (CLI) and the speed of execution of FastTrack.

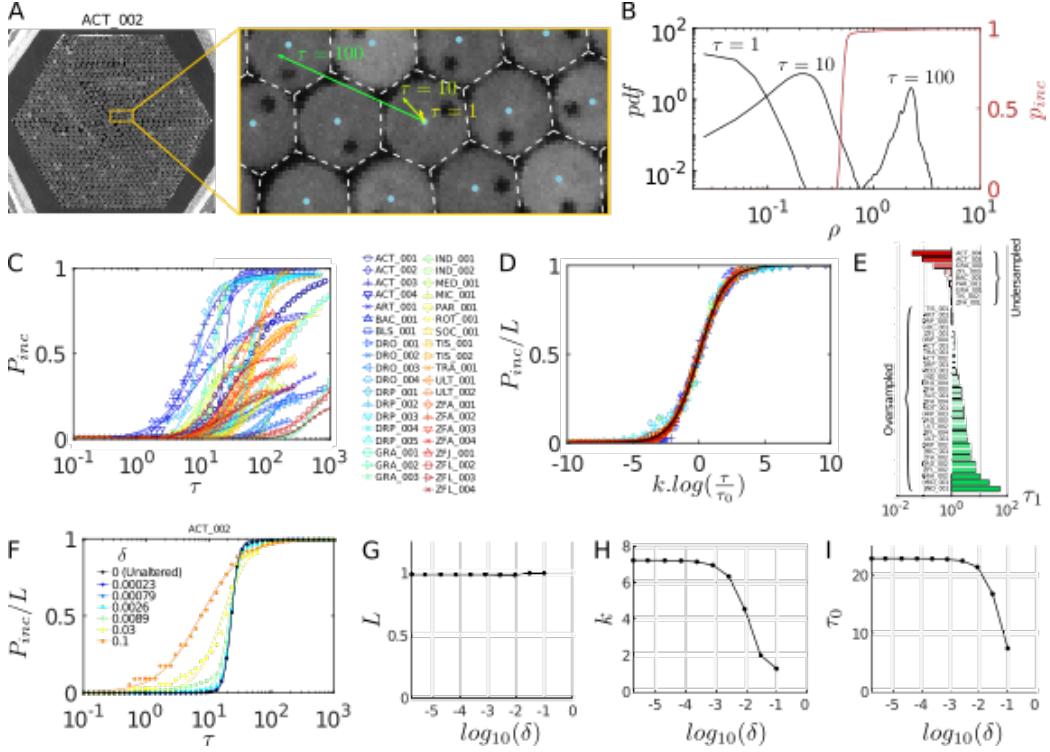


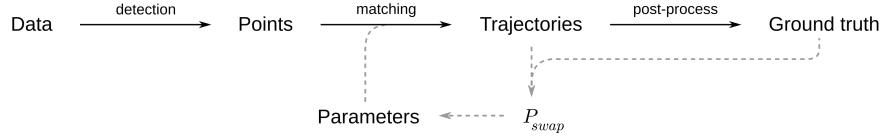
Figure 4.3: Characterization of the TD² dataset. **(A)** Illustration of the dynamics at various timescales in *ACT_002*. The Voronoï cells (dashed white) and the displacements of a particle at $\tau = 1, 10$ and 100 are overlaid. **(B)** Geometric probability of incursion p_{inc} (red) and distribution of the reduced displacement ρ at three different timescales τ (black) in *ACT_002*. The probability of incursion P_{inc} is the intersection of the areas under the two curves. **(C)** P_{inc} as a function of τ for the whole dataset (symbols). The solid lines are fits with a logistic function (see text). **(D)** Scaling of the reduced quantities P_{inc}/L as a function of $k.\log(\frac{\tau}{\tau_0})$ on the standard logistic sigmoid function (solid black). **(E)** Classification of the movies in the dataset by increasing values of τ_1 as defined by eq. (??), with fitting parameters determined over a logarithmic scale for P_{inc} . Movies with $\tau_1 < 1$ are undersampled while movies with $\tau_1 > 1$ are oversampled. **(F)** Comparison of $P_{inc}(\tau)$ for different levels of degradation δ (symbols) and corresponding logistic fits (solid curves) in *ACT_002*. **(G-I)** Evolutions of the fitting parameters L , k and τ_0 as a function of the degradation δ in *ACT_002*

Let us first apply this approach to gain insight into the influence of h_r , i.e., the maximal distance allowed for an object to travel before consid-

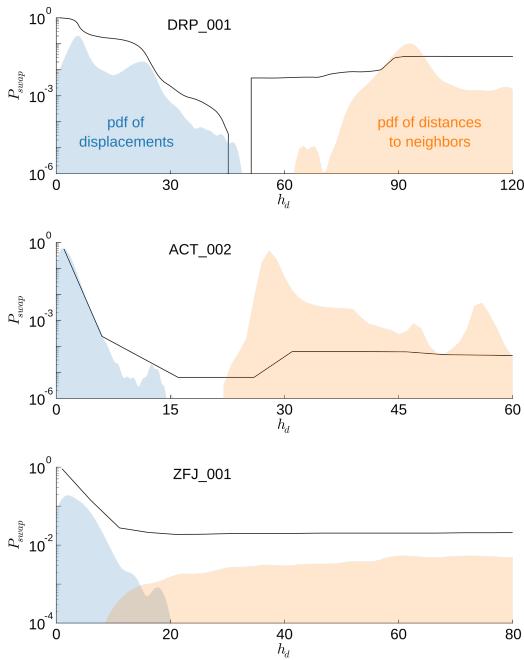
ered lost. The Figure 4.4-B displays how P_{swap} evolves as a function of h_r for three recordings in the dataset. For low values of h_r , P_{swap} is essentially imposed by the distribution of the objects' displacements since a high number of errors are generated when the objects are not allowed to move sufficiently. For higher values of h_r , the distribution of the distances to the neighbors (as defined by the Voronoï tesselation) starts to influence P_{swap} as the algorithm becomes sensitive to incursions. It can also be more easily fooled by entries and exits at the boundaries of the region of interest when the number of objects in the scene varies. In between, for most recordings, there is a gap yielding the minimal probability of error. That is particularly true when the objects are densely packed, since the distribution of distances to neighbors is sharper, like for *DRP_001* where P_{swap} drops to zero on a range of h_r . The acquisition framerate also has an essential role in this effect. With highly time-resolved movies, the distribution of displacements is shifted to the left (i.e., short distances), leading to a clear separation between the distribution of displacements and the distribution of the distances to the neighbors, resulting in low values of P_{swap} . In contrast, for poorly time-resolved movies like *ZFJ_001* the two distributions overlap, and P_{swap} is always bound to high values.

Similar analysis can be performed on the other tracking parameters. The Figure 4.4-C represents P_{swap} as a function of both hard parameters h_r and h_t for *PAR_001*, and a thin optimal segment appears. The Figure 4.4-D represents P_{swap} as a function of the two soft parameters s_r and s_α , and an optimal ratio lies at $\frac{s_r}{s_\alpha} \simeq 0.63$. Altogether, a set of optimal parameters can be derived and used for the processing of similar recordings.

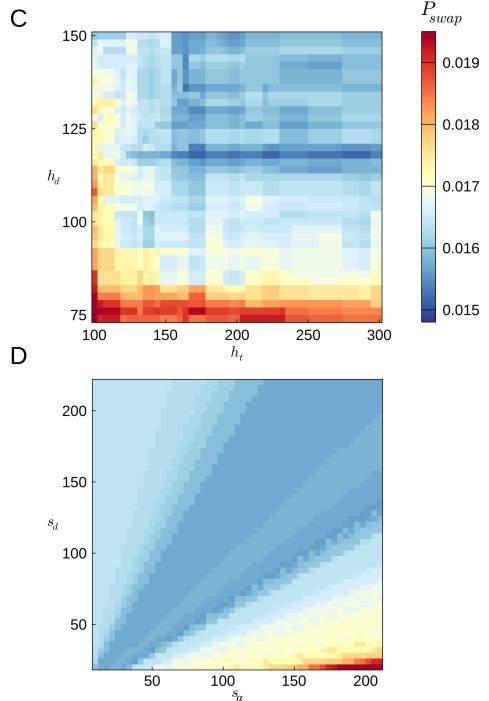
A



B



C



D

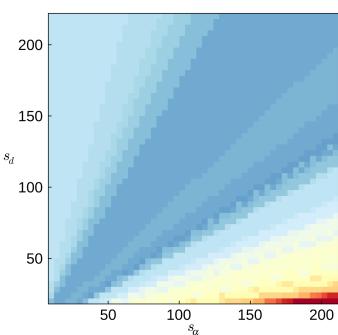


Figure 4.4: Optimization of tracking parameters based on P_{swap} . (A) Scheme of the optimization workflow: on top of the detection/matching/post-process flow chart, the ground truth is used to compute P_{swap} and create a feedback loop on the tracking parameters. (B) P_{swap} (black) as a function of the maximal distance parameter h_r (in pixels) for three typical recordings. Vertical lines for DRP_001 indicate that P_{swap} drops to 0. The distributions of displacements between successive frames (blue) and of distances to the neighbors (orange) are also shown for comparison. (C) P_{swap} as a function of the maximal distance parameter h_r (in pixels) and the maximal disappearance time h_t (in frames) for PAR_001 . Soft parameters are set to $s_r = 95$ and $s_\alpha = 60$. (D) P_{swap} as a function of the normalization distance parameter s_r (in pixels) and the normalization angle s_α (in degrees) for PAR_001 . Hard parameters are set to $h_r = 210$ and $h_t = 90$.

⁸⁹⁶ Chapter 5

⁸⁹⁷ Perspective

⁸⁹⁸ In these chapters, we saw how we implemented a versatile and easy
⁸⁹⁹ to use tracking software using open-source tools. Taking advantage of
⁹⁰⁰ the GitHub Actions system, we automated the testing and the deploy-
⁹⁰¹ ment of the software, increasing confidence, and promoting collaboration.
⁹⁰² We have shown that FastTrack can compete with state-of-the-art track-
⁹⁰³ ing software for many usages. At the same time, we compile a dataset
⁹⁰⁴ of movies, allowing us to benchmark tracking software on a wide variety
⁹⁰⁵ of systems. We classify the dataset based on the probability of incursion
⁹⁰⁶ and, doing so, highlight a criterion to determine the optimal framerate
⁹⁰⁷ of acquisition. We have finally shown how to determine the best set of
⁹⁰⁸ tracking parameters by leveraging FastTrack's full capabilities.

⁹⁰⁹ FastTrack's original approach, shifting the workload on the post-processing
⁹¹⁰ phase while keeping the pre-processing as light as possible, allows the
⁹¹¹ use of FastTrack without insight into the system to track. The post-processing
⁹¹² phase, mainly a swift checking of the tracking and small corrections, can
⁹¹³ be done directly inside the software in an interactive and ergonomic envi-
⁹¹⁴ ronment. FastTrack allows users to track movies quickly without any com-
⁹¹⁵ puter knowledge.

⁹¹⁶ FastTrack's approach does not prevent human inputs, mainly in the

917 post-processing phase, to obtain a perfect tracking accuracy. It will be
918 without inconvenience for many users who will need a human checking
919 in any case. However, users who want a perfect tracking accuracy without
920 human input will have to turn to other tracking software.

921 It is important to note that the source code of FastTrack is available
922 with a fully documented API. Power users can specialize the software with
923 a custom detection phase or a custom cost function tailored to their sys-
924 tem to circumvent any encountered limitation. The FastTrack command-
925 line interface allows to embed the software in a high-level programming
926 language like Python or Julia and integrate it inside an existing workflow.

927 Overall, FastTrack gives any user the power to quickly analyze their
928 movies on a relatively modest computer and power-user to build a custom-
929 tailored software. The feedback we have encountered more frequently is
930 how to analyze the tracking results. The standardized output leaves the
931 user free to choose the analysis tool that he preferred. An answer to this
932 request will be to develop analysis add-ons integrated into FastTrack if
933 needed. These add-ons will be thematic (e.g., rats behavior, soft matter,
934 etc.), and each one will have a specific set of functions to compute mean-
935 ingful quantities specific to this domain and system. Another perspective
936 that can be envisioned is to include the possibility of live tracking inside
937 the software.

Appendices

938

939 **Appendix A**

940 **Voronoi diagram**

941 **A.1 Definition**

942 The Voronoi diagram is a partition of a spatial plan containing points
943 into convex polygons, such as each polygon contains exactly one point.

944 **A.2 Construction**

945 In general position ¹, the dual graph of the Voronoi diagram is the De-
946 launay triangulation. The Delaunay triangulation is a triangulation where
947 every circumcircle is an empty circle. The circumcenters of Delaunay tri-
948 angles are the vertices of the Voronoi diagram.

¹An arrangement of points with no three collinear.

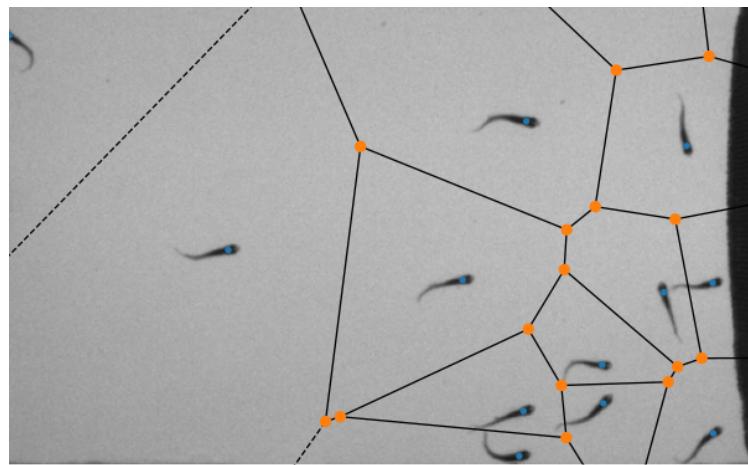


Figure A.1: Exemple of a Voronoï diagram computed with one image of ZFJ_001.
Voronoi vertices represented with orange points, seed points with blue points, finite Voronoï ridges with black lines, and infinite Voronoï ridges with dashed-black lines.

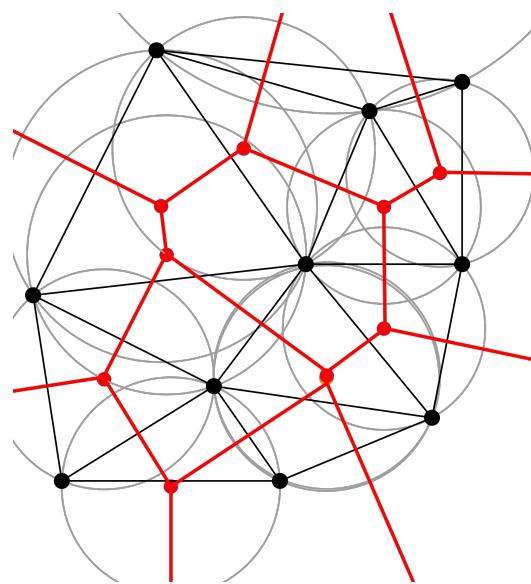


Figure A.2: Delaunay triangulation and Voronoï diagram. Delaunay triangulation in black, circumcircles in grey and Voronoï diagram in red.

949

Appendix B

950

Hungarian algorithm

951

B.1 Definition

952 The Hungarian algorithm is a combinatorial optimization problem that
953 solves the so-called assignment problem in polynomial time [30]. Since
954 1957, it has been known as the Kuhn–Munkres algorithm [31] after that
955 James Munkres reviewed it as strongly polynomial. First $O(n^4)$, several
956 implementations exist with a complexity of $O(n^3)$ [32–34].

957

B.2 Description

958 **Problem:** We consider four jobs J₁, J₂, and J₃ that need to be executed
959 by four workers W₁, W₂, and W₃, one job per worker. The objective is to
960 minimize the total cost ¹. In this example, we choose the simplest form of
961 the problem with a square matrix.

¹<http://www.hungarianalgorithm.com>

	$J1$	$J2$	$J3$	$J4$
$W1$	14	27	92	59
$W2$	38	43	50	17
$W3$	10	42	64	67
$W4$	88	32	83	89

Step 1: subtract the row minimum from each row:

$$\begin{array}{ccccc}
 0 & 13 & 78 & 45 & (-14) \\
 21 & 26 & 33 & 0 & (-17) \\
 0 & 32 & 54 & 57 & (-10) \\
 56 & 0 & 51 & 57 & (-32)
 \end{array}$$

Step 2: subtract the column minimum from each row:

$$\begin{array}{ccccc}
 0 & 13 & 45 & 45 & \\
 21 & 26 & 0 & 0 & \\
 0 & 32 & 21 & 57 & \\
 56 & 0 & 18 & 57 & \\
 (-0) & (-0) & (-33) & (-0) &
 \end{array}$$

Step 2: Covers all 0 with a minimum number of lines:

$$\begin{array}{ccccc}
 \textcolor{orange}{0} & 13 & 45 & 45 & \\
 \textcolor{orange}{21} & \textcolor{orange}{26} & \textcolor{orange}{0} & \textcolor{orange}{0} & x \\
 \textcolor{orange}{0} & 32 & 21 & 57 & \\
 \textcolor{orange}{56} & 0 & \textcolor{orange}{18} & \textcolor{orange}{57} & x \\
 & & x & &
 \end{array}$$

Step 4: Find the smallest element k not covered, subtract k to all uncovered elements and add k to all elements that are covered twice:

$$\begin{array}{ccccc}
 0 & 0 & 32 & 32 & \\
 34 & 26 & 0 & 0 & \\
 0 & 19 & 8 & 44 & \\
 69 & 0 & 18 & 57 &
 \end{array}$$

Repeat step 3 and 4 until there is exactly the same number of lines to covers all the 0 than the number of lines in the matrix. The optimal assignment is given by taking the set of 0 with one zero by line and by column, the cost by taking the value of these 0 in the initial matrix:

0	0	24	24
42	34	0	0
0	19	0	36
69	0	10	49

- 962 In this case the total cost is 127 with the assignment $\{J1; W1\}$, $\{J2; W4\}$,
963 $\{J3; W3\}$ and $\{J4; W2\}$.

⁹⁶⁴ **Appendix C**

⁹⁶⁵ **Displacement distribution fitting**

⁹⁶⁶ The standardized χ distribution with 2 degree of freedom reads:

$$f_0(x) = xe^{-\frac{x^2}{2}} \quad (\text{C.1})$$

⁹⁶⁷ with the mean $\mu_0 = \frac{\sqrt{2\pi}}{2}$ and the variance $\sigma_0^2 = 2 - \mu_0^2 = \frac{4-\pi}{2}$

The generalized χ distribution with 2 degree of freedom, A a shift and B a scaling reads:

$$f(x) = \frac{x - A}{B^2} e^{-\frac{1}{2}(\frac{x-A}{B})^2}$$

⁹⁶⁸ with the mean $\mu = \mu_0 B + A$ and the standard deviation $\sigma = \sigma_0 B$ with μ_0
⁹⁶⁹ and σ_0 the mean and standard deviation from the standardized χ distri-
⁹⁷⁰ bution. Substituting by $A = \mu - \mu_0 \frac{\sigma}{\sigma_0}$ and $B = \frac{\sigma}{\sigma_0}$ we obtain:

$$f(x) = \frac{x - \mu + \mu_0 \frac{\sigma}{\sigma_0}}{(\frac{\sigma}{\sigma_0})^2} e^{-\frac{1}{2}(\frac{x-\mu+\mu_0\frac{\sigma}{\sigma_0}}{\frac{\sigma}{\sigma_0}})^2} \quad (\text{C.2})$$

⁹⁷¹ We can approximate that $A = \mu - \mu_0 \frac{\sigma}{\sigma_0} = 0$ in the large majority of
⁹⁷² cases leading to:

$$f(x) = \frac{x}{(\frac{\sigma}{\sigma_0})^2} e^{-\frac{1}{2}(\frac{x}{\frac{\sigma}{\sigma_0}})^2} \quad (\text{C.3})$$

Part II

973

Dual

974

975 Chapter 6

976 Introduction

977 6.1 The chemical perception

978 The chemical senses are one the oldest sensory system [35]. They are
979 the most used sensory modality and observed in a wide range of taxa,
980 from unicellular [36] to mammalian. Some features and basics principles
981 are highly conserved across phyla [37, 38], and mediate several behav-
982 iors like predator avoidance, food-finding, and mating necessary to the
983 survival of any species.

984 Fish are immersed in their chemical environment at any time. This
985 chemical environment is rich in information, and fish have evolved a com-
986 plex sensory system to perceived and interpret these stimuli. For fish,
987 chemical perception is mediated by three organs: olfaction (smell), gus-
988 tation (taste), and a common chemical sense. Unlike terrestrial species,
989 where substances perceived by smell and taste differ by the medium
990 of transport of the molecules, fish taste and smell through the same
991 medium: water. The solubility of compounds in water determines the type
992 of compounds that can be transported and perceived. Therefore, chem-
993 ical perception is non-directional and persistent. The distance traveled

994 and the perceived concentration depend on the diffusion and convection
995 of the medium, determining the perception threshold and the compound's residence time in the environment. The chemical perception is
996 extremely specific, being contained in the molecular structures and the
997 complex mixture of chemicals.
998

999 Mechanisms of perception have been well studied for diverse fish species
1000 [39], but highly complex directed behaviors like homing migration or food-
1001 finding are still poorly understood [40]. For example, fish can find food in
1002 complex environments like turbid and turbulent waters, where the per-
1003ception is fragmented. Deciphering these mechanisms and their associ-
1004 ated neural mechanisms will significantly advance the comprehension of
1005 the animal kingdom's most used sensory modality.

1006 One fish species, the zebrafish is an emerging model for studying goal-
1007 driven behaviors. At the larval stage (6 days post-fertilization), the animal
1008 is transparent, and it is possible to observe the brain activity with cellular
1009 resolution using light-sheet microscopy [41]. The development of virtual
1010 reality assays makes it technically possible to associate some neuronal
1011 networks' activity to the observed behavior. Using this technique; it was
1012 possible to gain insights into several behaviors, such as prey capture [42],
1013 optomotor response [43], phototaxis [44], rheotaxis [45], and thermotaxis
1014 [46].

1015 The development and functioning of the fish sensory organs, partic-
1016 ularly in the zebrafish, have been well characterized. However, there are
1017 few behavioral studies on chemical perception and chemically-oriented
1018 navigation. Several milestones have to be achieved before using virtual
1019 reality assays to study chemically-driven navigation. One needs to find
1020 and characterize products that elicit robust and attractive behaviors. The
1021 space of possibilities will be vast, and completing this task necessitates
1022 a high-throughput setup to explore combinations of products, concen-
1023 trations, and fish ages. In a second time, when a product that elicits a
1024 robust and attractive response will be characterized, studying realistic
1025 chemically-driven navigation will require a setup capable of reproducing

1026 turbulent flows where the fish is immersed in a complex chemical envi-
1027 ronment with fragmented perceptions.

1028 In the next sections, we will present in detail the sensory organs of the
1029 zebrafish and review experimental setups used to characterize the chem-
1030 ical perception of the zebrafish at the larval and adult stage. Then we will
1031 present two experimental setups that we build: Dual, a high-throughput
1032 screening device capable of assessing the chemical preference of larval
1033 and adult zebrafish; The Tropical River, a setup capable of generating con-
1034 trolled flows that can be used to study chemically-driven navigation. Fi-
1035 nally, we will present some results that we obtain using the Dual setup.

1036 6.1.1 Olfaction

1037 The olfactory organ of the fish Figure 6.1 consists of two structures
1038 located in the animal's snout. Each structure consists of a cavity called
1039 the olfactory chamber connected to the outside by an entrance and an
1040 exit nostril. The inside of the olfactory chamber is lined with the olfac-
1041 tory rosette consisting of two rows of olfactory lamellae. The olfactory
1042 epithelium, where the olfactory receptors are located, is placed on these
1043 lamellae. The olfactory organ's exact organization and position can vary
1044 depending on the fish species, for example, with the addition of a venti-
1045 lation cavity as an extension of the olfactory cavity.

1046 The olfactory epithelium has a $100\mu m$ thick stratified columnar struc-
1047 ture. It can be separated into a sensory and a non-sensory epithelium.
1048 The sensory epithelium consists of three types of cells: receptor, support-
1049 ing, and basal cells; the non-sensory epithelium of goblet cells and non-
1050 sensory ciliated cells. There are five receptor cells implicated in the olfac-
1051 tory perception: ciliated cells, microvillous cells, crypt cells [47,48], kappe
1052 cells [49], and pear-shaped cells [50]. They express olfactory receptors of
1053 the OR, V1R, V2R, and TAAR families. Receptor cells have various sizes,
1054 shapes, and distribution inside the epithelium see Figure 6.2.

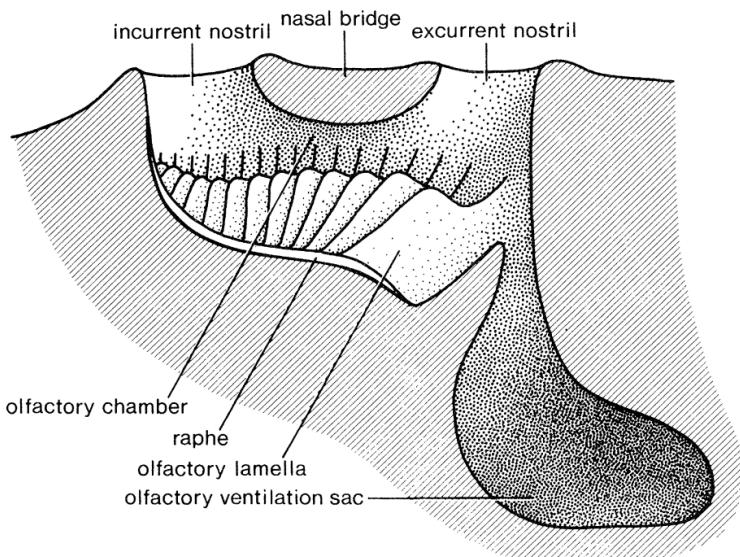


Figure 6.1: Olfactory system, reproduced from [39].

1055 Receptor cells project directly into the olfactory bulb located in the
 1056 brain, in turn sending signals to the telencephalon and diencephalon [51].
 1057 The olfactory bulb in the teleost is a structure of four concentric layers:
 1058 olfactory nerve layer (ONL), glomerular layer (GL), mitral cell layer (MCL),
 1059 and internal cell layer (ICL). The olfactory information is transmitted by
 1060 the receptor cells to the olfactory bulb [52] then in the forebrain [53] as a
 1061 topographical odor map. The olfactory bulb's neuronal connections have
 1062 been particularly studied in the zebrafish [54,55], the olfactory bulb com-
 1063 prised approximately 20 000 neurons [56] and 140 glomeruli [57]. Each
 1064 receptor cell expresses only one type of olfactory receptor [58–61] except
 1065 in a subpopulation of olfactory sensory neurons [61]. Cells expressing
 1066 the same receptor are projecting into the same olfactory bulb glomeruli
 1067 [62]. Glomeruli responding to similar odorants are grouped into domains
 1068 within the olfactory bulb, forming chemotopic maps. Odorants can acti-
 1069 vate glomeruli outside their domain, leading to a fragmented map inside
 1070 the olfactory bulb [63]. Moreover, the odor encoding is hierachized with
 1071 first-order features encoded by large domains and second-order features
 1072 by local activity patterns within the domain [64,65].

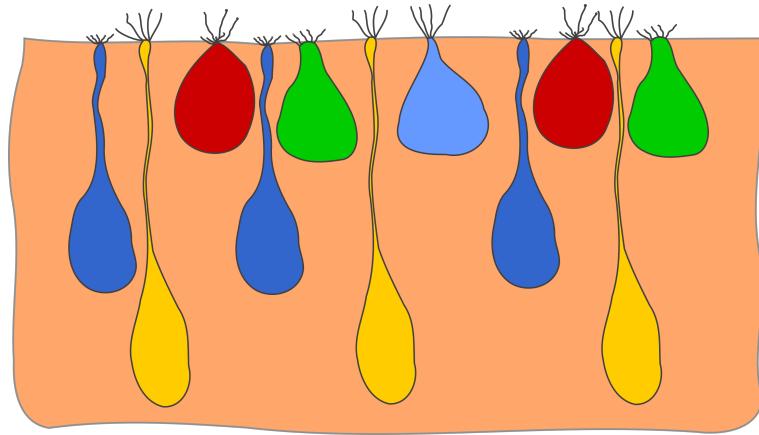


Figure 6.2: Schematic representation of the olfactory epithelium. Ciliated neuron in yellow with round somata, slender dendrite, and cilia. Microvillous neuron in dark blue with microvilli at the surface. Crypt neuron in red with ovoid shape, microvilli, and cilia. Kappe neuron in green with microvilli. Pear-shaped neuron in light blue with cilia.

1073 The olfactory bulb projects into two higher brain structures, the telencephalon (Dp and Vv) and the diencephalon (habenula, posterior tubercle, and hypothalamus). Odor response in these areas is currently poorly
1074 understood.
1075
1076

1077 In the zebrafish [66, 67], the olfactory organ develops from the olfactory placodes at the 6-10 somites stage (about 15 hours post-fertilization)
1078 of the embryonic development. The olfactory cavity begins to appear at
1079 the 28-30 somites stage (31 hours post-fertilization). Approximately 50
1080 hours post-fertilization, the olfactory epithelium and the receptor cells
1081 appear. When the embryo emerges from the egg, 4 days post-fertilization,
1082 the olfactory organ continues its morphological development, but the
1083 cytological organization changes little. At 40 days post-fertilization, the
1084 bridge between the entrance nostril and the exit nostril is completely
1085 formed, separating the currents going out and coming in from the olfac-
1086 tory cavity. The addition of lamellae to the olfactory rosette continues
1087 throughout the life of the zebrafish.
1088

1089 **6.1.2 Gustation**

1090 The gustatory organ of fish consists of the taste buds, which directly
1091 contact chemical substances. Taste bud histology has been studied for
1092 different fish [68–72]. They usually have an elongated and ovoid shape.
1093 They sit on a small dermal papilla and extend throughout the epidermis'
1094 thickness protruding from the surface. The taste bud is constituted of a
1095 sensory (dark cells with microvilli and light cells with one large microvil-
1096 lus) and a non-sensory (Merkel-like basal cells) epithelium. The apical
1097 ending of the sensory cells that protrude from the epithelium is called
1098 the receptor field and covers with a mucous cap. The number of sensory
1099 cells in a taste bud varies considerably depending on the fish species.

1100 Taste buds are distributed all over the fish's body, especially in the
1101 mouth, on the lips, and the skin. Their distribution and concentration vary
1102 according to the species. Three different cranial nerves innervate them: fa-
1103 cial (VII), glossopharyngeal (IX), and vagal (X). The facial nerve transmits
1104 information from the extra-oral taste buds; the glossopharyngeal nerve
1105 transmits information from inside the oral cavity; the vagal nerve trans-
1106 mits information from inside the oropharyngeal cavity. The taste system
1107 is anatomically divided into two distinct parts: nerves IX and X projecting
1108 into the brain's vagal lobe and nerve IV into the facial lobe. Connections
1109 to higher areas of the brain differ slightly from one species to another. It
1110 has been shown in *Ictalurus nebulosus* [?] that these two systems have
1111 distinct roles in fish feeding behavior.

1112 In zebrafish [74], the taste buds (approximately 200) are located on
1113 the lips, in the oropharyngeal cavity on the barbels, and on the head's ven-
1114 tral and dorsal side. Each taste bud contains 20 to 23 cells. Projections of
1115 the zebrafish gustatory system have been studied in detail [75] and form
1116 a complex network that can be summarized graphically see Figure 6.6.

1117 The development of the gustatory organ has been studied in the ze-
1118 brafish [73]. The first taste buds appear at 3 or 4 days post-fertilization and

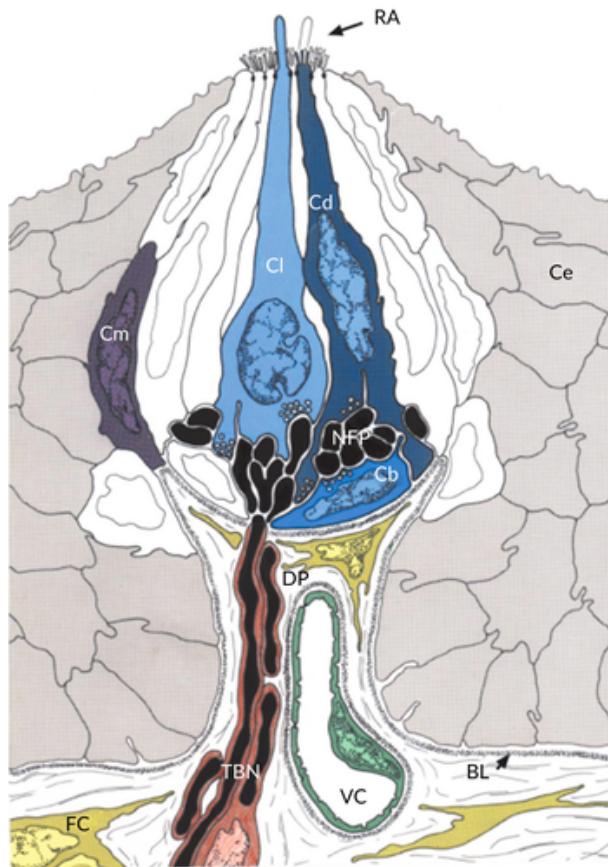


Figure 6.3: Schematic drawing of a typical taste bud of teleosts from [73]. Dark cells (Cd), light cells (Cl) and Merkel-like basal cells (Cb). Marginal cells (Cm). Ce epithelial cells. Dermal papilla (DP). (TBN) taste bud nerve. (BL) basal lamina. (RA) receptor area. (VC) capillary vessel.

1119 are located on the lips and the gill arches. The taste buds in the mouth
 1120 and oropharyngeal cavity appear 4 to 5 days post-fertilization. The taste
 1121 buds on the head do not appear until 12 days post-fertilization, and it is
 1122 not until the juvenile stage (30 to 40 days post-fertilization) that the bar-
 1123 bels appear. Note that the appearance of the taste buds coincides with
 1124 the appearance of feeding in the larvae.

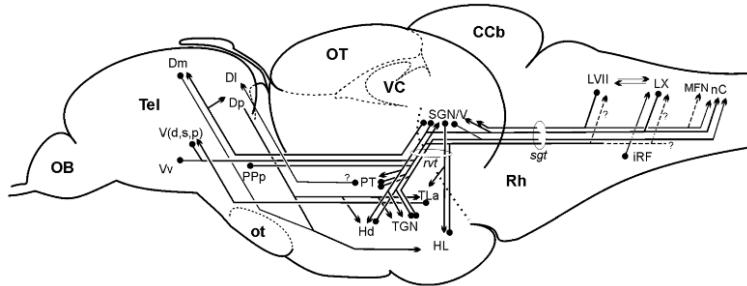


Figure 6.4: Gustatory system of the zebrafish. Neuronal connections of the fish taste system reproduced from [75].

1125 6.1.3 Common chemical sense

1126 Fish also have a third chemical sense called the common chemical
 1127 sense. It consists of bipolar neurons called solitary chemosensory cells
 1128 (SCCs) embedded in the epidermis. Their distribution and number vary
 1129 greatly depending on the species. Therefore their study is difficult, and
 1130 their function and neuronal connections are poorly understood.

1131 In the zebrafish [76], SCCs have been described as a set of 2-7 villi of
 1132 0.5 to 1 μm length emerging from the cell body at embryo and larval stage.
 1133 In adults, SCCs possess a single villus of 3 μm length.

1134 The first SCCs appear at 3 days post-fertilization. Their density increases
 1135 until 25 days, where their number stabilizes at 1.10^6 per mm^2 with 2 to 5
 1136 times more SCCs on the zebrafish's head than on its body.

1137 6.1.4 Behavior

1138 The olfaction and gustation have been shown to mediate several fish
 1139 behaviors. It is not easy to distinguish the contribution of each sense in
 1140 the observed behavior. Moreover, this contribution seems to be depen-
 1141 dent on fish species.

1142 A well known and impressive behavior encountered in many fish species
1143 is the homing migration. A typical example is salmons that perform three
1144 migratory phases throughout their life. One of them, the upstream mi-
1145 gration from the ocean to their home stream, has been shown to rely
1146 on an olfaction imprinting [77, 78]. Little is known about the imprinting
1147 mechanism, but experiments suggest that it relies on a mixture of odors
1148 perceived during the juvenile stage in the fish's home stream.

1149 Feeding is one of the most important behaviors. It relies on several
1150 senses for food detection, and selection [79]. A stereotyped behavioral se-
1151 quence was shown to exist [80] consisting in a step of arousal mainly me-
1152 diated by olfaction [81], then a step of localization of the food mediated
1153 by chemical and visual cues. The last step of ingestion is triggered pri-
1154 marily by the gustation [80]. The impact of each sensory modality varies
1155 significantly with the species. For example, the yellow bullhead has the
1156 entire feeding sequence mediated by taste, whereas ictalurid catfish prey
1157 detection was abolished when olfaction was blocked. The chemical sub-
1158 stances that attract fish depend on the species [80], and response to a
1159 mixture is higher than isolated compounds in general.

1160 Olfaction [82] as well as the gustatory system [83] has been shown to
1161 play an essential role in reproduction. Non-anosmic males exposed to
1162 water taken from a tank with a gravid female developed courtship behav-
1163 iors, except for some species like the three-spined stickleback where the
1164 gustation can replace the olfaction. Complete courtship repertoire neces-
1165 sitated the presence of other sensory cues.

1166 Fright reaction occurred when a fish perceived an alarm substance se-
1167 creted by a conspecific. This reaction differs between species and involves
1168 seeking cover, rapid swimming, or freezing. It is accepted to be mediated
1169 by olfaction [84–86], but other sensory cues are not ruled out.

¹¹⁷⁰ **6.2 Behavioral studies**

¹¹⁷¹ The zebrafish has been used to connect the behavioral and neuronal
¹¹⁷² response to diverse stimuli: visual stimuli [], temperature [], and balance
¹¹⁷³ reflexes []. Most of the works to date focus on developing the model for
¹¹⁷⁴ pharmacological safety screening [87], drugs addiction [88], and ecology
¹¹⁷⁵ [89], enabling a low cost and genetically manipulable model. Behavioral
¹¹⁷⁶ studies of the chemical perception of zebrafish, adults, or at the larval
¹¹⁷⁷ stage have been done through various experimental apparatuses that we
¹¹⁷⁸ will present in the following sections.

¹¹⁷⁹ **6.2.1 Conditioned place preference**

¹¹⁸⁰ The conditional place preference (CPP) experiment is a type of Pavlo-
¹¹⁸¹ vian conditioning. Pavlovian conditioning consists of associating a condi-
¹¹⁸² tioned stimulus (generally neutral) with an unconditioned stimulus. After
¹¹⁸³ learning, the animal exhibits a conditioned response to the conditioned
¹¹⁸⁴ stimulus when presented alone. The most classic example is associating
¹¹⁸⁵ a bell's sound (conditioned stimulus) to the release of a food smell (un-
¹¹⁸⁶ conditioned stimulus). After learning, the animal can respond to the bell's
¹¹⁸⁷ sound alone.

¹¹⁸⁸ This approach was applied to test the response to various chemical
¹¹⁸⁹ stimuli in adult zebrafish [90]. The experiment follows a classical 3-step
¹¹⁹⁰ design. The first step is to evaluate the fish's base-line preference. The
¹¹⁹¹ animal is placed in an aquarium with two or three distinct areas differ-
¹¹⁹² entiate by walls' pattern and color, see Figure 6.5. The fish is tested to
¹¹⁹³ find out which side it naturally prefers. In this experiment, the distinctive
¹¹⁹⁴ walls' pattern and color play the role of the conditioned stimulus. The
¹¹⁹⁵ second step is the conditioning phase. The fish is restrained to its least
¹¹⁹⁶ preferred area, and the substance to be tested injected into the water
¹¹⁹⁷ (unconditioned stimuli). The last step consists of repeating the first step

₁₁₉₈ to assess the change in preference of the animal.

₁₁₉₉ Several chemical substances have been tested using this method [91–
₁₂₀₀ 93]. Notably, a strong and robust cocaine-induced CPP response in WT
₁₂₀₁ zebrafish was shown [94], with 85% of the fish changing preference to a
₁₂₀₂ cocaine concentration of 10mg.L^{-1} and lower and higher concentrations
₁₂₀₃ resulting in a lower response. A positive response of adult zebrafish to a
₁₂₀₄ single ethanol exposure was shown [95] in a similar experimental setup.
₁₂₀₅ It should be noted that this is also the first study to use an automated
₁₂₀₆ tracking system to calculate animal preference. Zebrafish showed a pos-
₁₂₀₇ tive response for D-amphetamine [96, 97], salvinorin A [98], cocaine [98],
₁₂₀₈ spiradoline [98], nicotine [99] and ethanol [99].

₁₂₀₉ We see that CPP has been used extensively to study the response to
₁₂₁₀ chemical stimuli in zebrafish. There is a strong emphasis on products that
₁₂₁₁ cause addictive pathologies in humans. Nevertheless, this protocol has
₁₂₁₂ several flaws, the first being that it involves several systems of perception
₁₂₁₃ as well as memory. During the conditioning phase, the learning is based
₁₂₁₄ on the visual perception of the environment (pattern on the aquarium
₁₂₁₅ walls), the chemical perception of the tested compound, the association
₁₂₁₆ of the two stimuli coming from different sensory organs, and the memo-
₁₂₁₇ rization of these perceptions. Secondly, the time window to perform the
₁₂₁₈ experiment (minimum two days) is a hindrance to use the CPP to study
₁₂₁₉ the effect of many chemicals in a high-throughput manner.

₁₂₂₀ 6.2.2 Well-plate

₁₂₂₁ A widely used experimental apparatus to assess chemical compounds'
₁₂₂₂ effect on zebrafish larvae and embryos is the well-plate device [100]. One
₁₂₂₃ or more larvae is placed in each well in a bath of a chemical. Larvae are
₁₂₂₄ then recorded swimming in the chemical compound, and the kinematic
₁₂₂₅ parameters of the animal are extracted. In the case of embryos, devel-
₁₂₂₆ opment is monitored after exposure. The advantage of this technique is



Figure 6.5: Conditioned place preference apparatus. CPP setup for zebrafish reproduced from Brennan Caroline, Queen Mary University of London. The middle wall is removed for the first and third step of the CPP.

1227 that it requires little equipment. It quickly produces a large amount of
1228 data with well-plate up to 48 wells per plate. Software already exist to ex-
1229 tract automatically relevant behavior parameters from video recordings.

1230 With this device, many chemical compounds have been tested [101–
1231 103], as well as seizure liability [104], and several behaviors [105–108].

1232 The well-plate device allows an easy and automatic high-throughput
1233 screening of chemicals. Turnkey commercial solutions like the Zebrabox
1234 from ViewPoint exist, and custom setups are relatively easy to build. How-
1235 ever, this system suffers limitations like the fact that one can not as-
1236 sess the fish preference. Precisely controlled exposure, or repeated expo-
1237 sure through cycles of exposition/flushing, are not available. Therefore
1238 this system is not adapted to investigate fish's chemical preference and
1239 chemical-driven navigation.



Figure 6.6: A Zebrabox from Viewpoint. Zebrabox, the most used solution for well-plates experiments.

1240 6.2.3 Diffusion

1241 Some authors have tried to quantify chemically induced behavior by
1242 introducing a chemical compound directly into the tank and looking at
1243 the percentage of time spent close to the source. Notably, an attraction
1244 concentration-dependent to adenosine for adult zebrafish [50] and to
1245 GCDA and nicotine for zebrafish larval [109] was shown. A strong aversion
1246 to cadaverine, an odor associated with decomposing bodies, was shown
1247 using a tank with a single compartment or a tank with two compartments
1248 and an intermediate zone where the fish can changes compartment [110],
1249 see Figure 6.7.

1250 Very easy to implement, these types of experimental devices lack con-
1251 trol in the concentration perceived by the animal. Diffusion and convec-
1252 tion are neglected in the experiment, and the concentration is poorly
1253 known and not reproducible. The effect of diffusion and convection was
1254 mitigated [110] by adding a wall separating the two zones Figure 6.7.F, al-

ways leaving uncertainty in the intermediate zone. Moreover, these setups exclude the realization of long experiments due to the homogenization of the product.

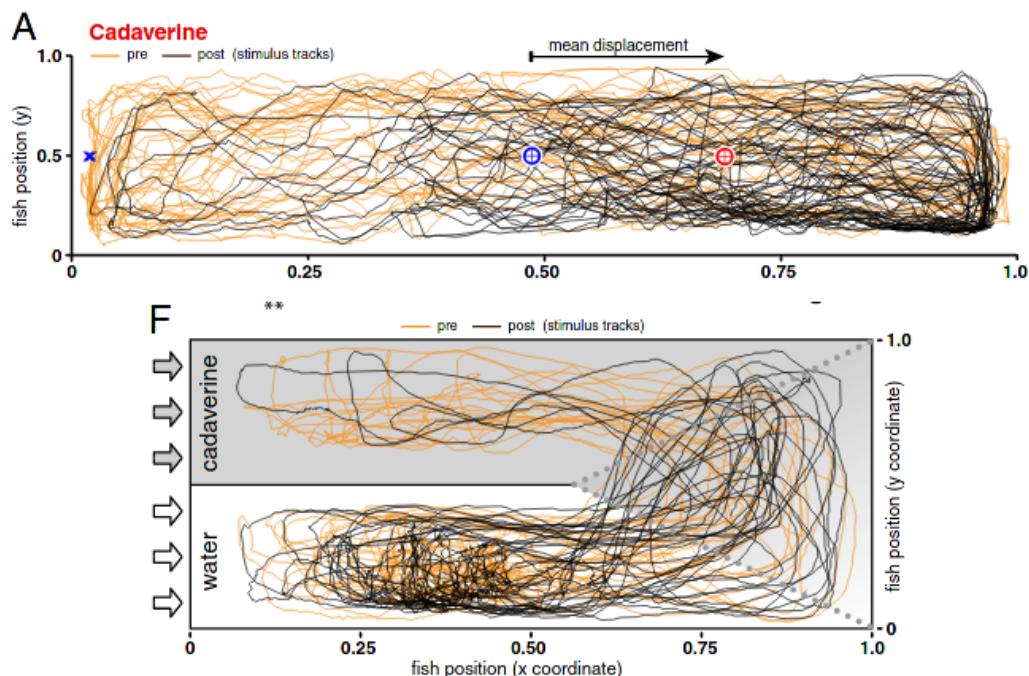


Figure 6.7: Diffusion setups from [110]. **A.** One channel diffusion setup, blue cross: chemical introduction point. **F.** Two channels setup.

6.2.4 Flow

An exciting setup to study chemical preference in fish is the underflow device. The first mention of this type of device date back to 2013 [111]. In this setup, the tank is separated into two distinct compartments using a laminar flow, see Figure 6.8. The animal can then choose between the two compartments during the experiment without any constraint, and the experimenter can put a chemical to test on one side. The interface between the two compartments self-heal with a characteristic time depending on the flow velocity. The time spent on each side, the number of interface

¹²⁶⁷ crosses, and the animal's kinematic parameter are extracted from video
¹²⁶⁸ recordings to assess the fish's preference.

¹²⁶⁹ Several psychoactive substances have been tested on adult zebrafish
¹²⁷⁰ [112, 113] and showed attraction by diazepam, fluoxetine, risperidone, and
¹²⁷¹ buspirone; neutral response to ethanol and clonazepam; an aversion to
¹²⁷² acid pH, two food odor extracts, and conditioned water took from a tank
¹²⁷³ with chemically and physically stressed fish.

¹²⁷⁴ This setup has several advantages. The concentration of the product
¹²⁷⁵ is perfectly known because the diffusion and convection are prevented
¹²⁷⁶ by the flow. The preference of the fish can be directly assessed because
¹²⁷⁷ the fish can choose freely to go inside or outside the product. Long ex-
¹²⁷⁸ periments can be performed with this setup, and product delivery pre-
¹²⁷⁹ cisely controlled in time. However, some disadvantages remain, like the
¹²⁸⁰ absence of a standardized or turnkey setup and the volume of water and
¹²⁸¹ chemicals required that can be high.

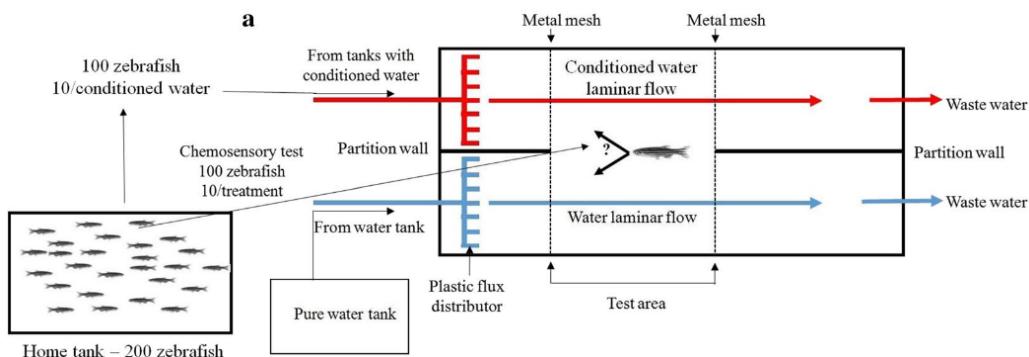


Figure 6.8: Flow setups from [113].

¹²⁸² Another type of flow device that allows the product's concentration
¹²⁸³ inside the tank to be quickly changed was used on adult zebrafish [114].
¹²⁸⁴ Like in the multi-well experiment, chemically induced behavior changes
¹²⁸⁵ were monitored by the animal's kinematic parameters.

¹²⁸⁶ Several food odors were shown to produce a significant increase in
¹²⁸⁷ speed and number of bursts; social odors from conspecific produced a

1288 similar response; alert odors result in a dive to the bottom of the tank and
1289 an increase in frozen time; decomposition odors result in more turns. The
1290 critical points noted with this device is the inter-and intra-experimental
1291 variability. The authors showed that less than a third of the odors used in
1292 the study produce reproducible results between trials of the same individ-
1293 ual. Some odors such as cadaverine, blood, skin, and food odors resulted
1294 in inconsistent responses for the same individual. Most odors produce
1295 poorly reproducible results for different fish.

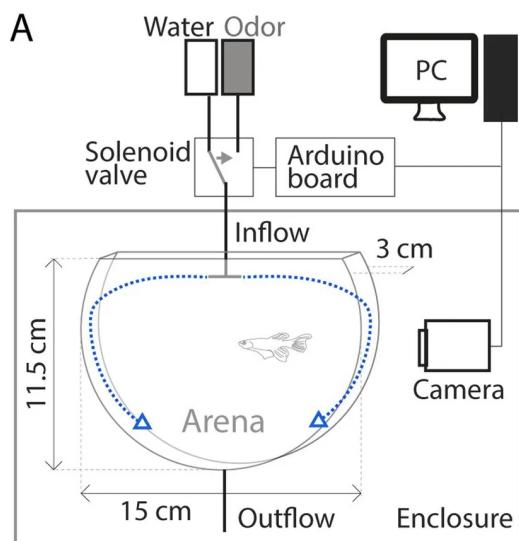


Figure 6.9: Flow setups from [114].

1296 This setup novelty is to allow adult animals to evolve realistically in
1297 a 3D environment and to have a better knowledge of the concentration
1298 perceived by the animal than in a diffusion setup. Therefore, direct prefer-
1299 ence assessment is not accessible, and comparisons with existing quasi-
1300 2D setups like the well-plate are difficult.

1301 From this overview of the scientific literature, we see that the study
1302 of chemical perception and behavioral response to chemical stimuli is
1303 not standardized. No experimental setup is set as a standard, and direct
1304 comparisons between studies carried out in various independent labo-
1305 ratories are not easily possible. In this context, we have developed Dual,
1306 an open-source, low cost, do it yourself, and scalable experimental setup.

₁₃₀₇ Using the underflow principle, Dual allows studying chemical preferences
₁₃₀₈ in larvae and juvenile zebrafish in a standardized, high-throughput, and
₁₃₀₉ comparable way.

1310 **Chapter 7**

1311 **Experimental setups**

1312 **7.1 Dual**

1313 The design and implementation of experimental setups that are high-
1314 throughput, bias-free, and scalable are essential to behavior character-
1315 ization. We present Dual, a high-throughput experimental setup that is
1316 easy to build, scalable, and costs less than 1 500 USD.

1317 **7.1.1 Overview**

1318 Dual is an underflow system builds on the same principle that [111]. It
1319 consists of creating two virtual compartments in a tank through a laminar
1320 flow. As we have seen previously, this system allows a rigorous knowledge
1321 of the compound's concentration to which the animal is subjected. Diffu-
1322 sion and advection due to the water movements caused by the fish are
1323 avoided. The interface between the two compartments is well defined
1324 and self-heal, when disturbed, with a characteristic time depending on
1325 the flow velocity.

1326 To create the laminar flow, Dual uses a system of four syringes, cou-
1327 pled two by two, in phases opposition: when two syringes push, two sy-
1328 ringes pull, see Figure 7.1. The syringes are connected to a millifluidic chip
1329 and generate a laminar flow with constant volume by having two syringes
1330 injecting at one side and two aspirating at the other side. Thanks to a mi-
1331 crovalves system, each syringe can be filled independently. One can then
1332 build an experimental protocol by stacking several filling and injection cy-
1333 cles and choosing what to fill the syringes with. The fish is placed inside
1334 the microfluidic chip, see Figure 7.4.

1335 For example, to assess fish preference, one can first create a cycle with
1336 water on the two sides as habituation and control. Then add a cycle with
1337 a chemical on one side and water on the other side to study the fish
1338 preference. Another example is to fill the two sides with a chemical for a
1339 given amount of time and then clean the system with water on the two
1340 sides, reproducing the type of experiment performed with the multi well-
1341 plate device. A large variety of experiments can be designed with none or
1342 few setup modifications.

1343 Dual is a Do It Yourself project, which allows great flexibility in the con-
1344 ception and fast iteration to adapt the project if necessary. All the compo-
1345 nents, blueprints, and other CAD files needed to build and assemble Dual
1346 are available in the appendixes. Dual can be built at a low-cost (see the
1347 bill of materials ??) without prior knowledge of mechanics or electron-
1348 ics. The tools necessary for realizing Dual can be found in a FabLab and
1349 necessitate little formation.

1350 **7.1.2 Construction**

1351 For the construction, Dual can be separated into five inter-dependent
1352 parts. The mechanical system comprised all the structures, the motor, and
1353 moving parts. The millifluidic system is constituted of the millifluidic chip,
1354 the microvalves, the syringes, and connecting tubes. A camera and LEDs

1355 form the imaging system. The electronic system is a custom PCB controlling
1356 the motor and the microvalves. Finally, a software that controls all
1357 the setup automatically.

1358 **Mechanical**

1359 Dual's central mechanical part is a motorized syringe push-pull pump
1360 that allows creating the laminar flow. It is built around the V-Slot Linear
1361 Actuator from OpenBuilds fixed on a structure builds using OpenBuilds
1362 linear rails and 3D printing fixations. A stepper motor drives the actuator,
1363 and two stop microswitches limit the range of motion, see Figure 7.1.

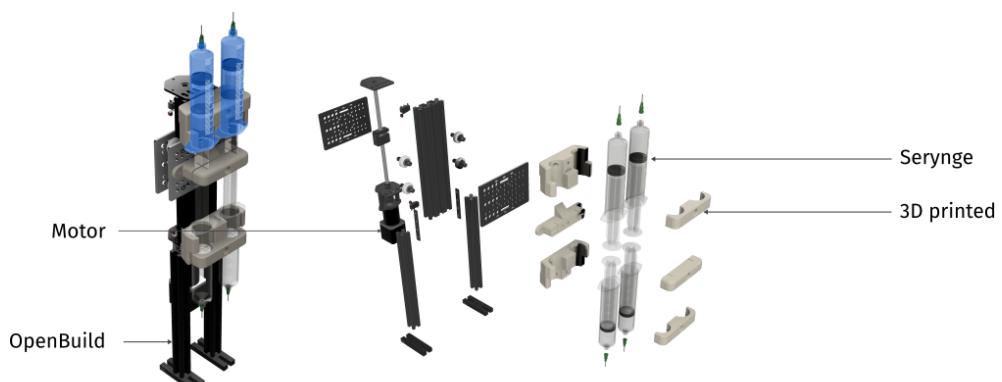


Figure 7.1: Dual mechanical structure Dual custom pull-push syringe.

1364 It is crucial to isolate the animal from the exterior environment and of
1365 any light sources. A box that will contain the millifluidic chip and lighting
1366 system is constructed using MakerBeam rails and medium-density fibre-
1367 board sheets. A PMMA infrared transparent sheet is placed on the remov-
1368 able top panel to record the experiment while blocking visible light. The
1369 box contains, see Figure 7.2, two LEDs for visible and infrared light, a dif-
1370 fuser for homogeneous lighting, and a support to fix the millifluidic chip.

1371 A structure to maintain the camera on top of the box and secure the
1372 Raspberry and the power alimentation is build using OpenBuilds rails. It
1373 is best to build Dual on top of an optical breadboard to facilitate fixation

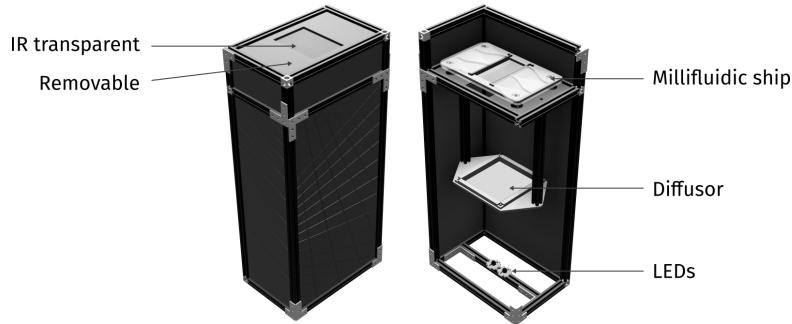


Figure 7.2: Box Dual box that will isolate the animal from the exterior.

1374 and enhanced stability. All these elements need to be fixed firmly and
 1375 level to avoid any bias that could disturb the animal during the experi-
 1376 ment.

1377 **Millifluidic**

1378 The millifluidic chip that will serve as a tank for the fish, see Figure 7.4
 1379 is laser cutted in PMMA plastic that is transparent and presents excellent
 1380 optic properties. The different parts are glued using acetic acid. The fish
 1381 is restrained in the center by nets, and profiled inputs and outputs allow
 1382 a laminar flow.

1383 Sixty-milliliters syringes are fixed with 3D printed fixations on the ac-
 1384 tuator and connected with 2.2 mm diameters tubing to the microvalves
 1385 circuit. Six three ports microvalves are connected and form a circuit that
 1386 allows performing cycles of filling and injection.

1387 **Imaging**

1388 The setup is lightened by transmission using an infrared LED placed
 1389 at the bottom of the box. Homogenous lighting is obtained by placing a
 1390 diffusor (tracing papers or white plastic sheet) at the box's mid-height.

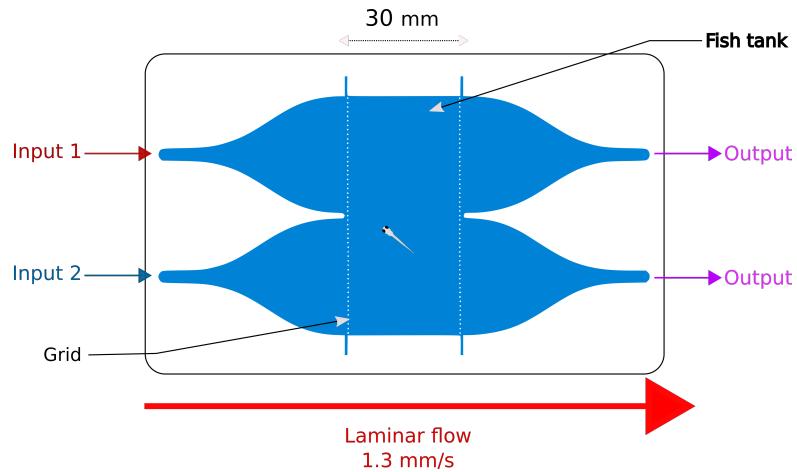


Figure 7.3:

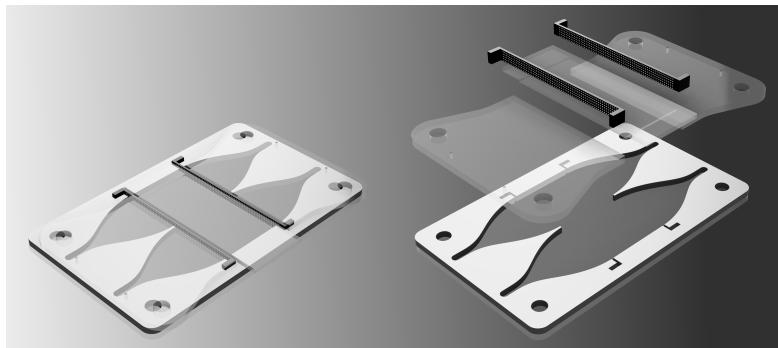


Figure 7.4:

¹³⁹¹ Experiments are recorded with a Chameleon 3 camera mounted with a
¹³⁹² 23-25 mm EFL lens connected directly to the computer via USB3. A PMMA
¹³⁹³ infrared transparent sheet is placed between the camera and the lens to
¹³⁹⁴ block visible light and only retrieve Dual's infrared lightening.

¹³⁹⁵ **Electronic**

¹³⁹⁶ The electronic system links the analogic mechanical and millifluidic
¹³⁹⁷ system to the software. A custom printed circuit board (PCB) has been
¹³⁹⁸ designed for Dual TODO(ajouter blueprint). It contains an Arduino Nano
¹³⁹⁹ serving as a microcontroller; six half-bridge gate drivers to control the

1400 microvalves from the logic signals of the Arduino; a Big EasyDriver stepper
1401 motor controller to control the motor speed from a logic output of the
1402 Arduino; a potentiometer to control the intensity of the LED lighting. All
1403 the electronics and the syringe pump motor are powered by a 550W ATX
1404 computer power supply connected to the PCB.

1405 **Software**

1406 A software has been specifically developed to control the setup. The
1407 graphical user interface is developed using Qt, and the camera is inter-
1408 faced to the software using the FLIR Systems SDK provided with the cam-
1409 era. The electronic system is controlled through the Arduino Nano flashed
1410 with a custom sketch and communicating to the software via USB serial.
1411 The software allows manual control of each system element, such as the
1412 microvalves, the camera, and the motor. It is possible (and advisable) to
1413 create custom experimental protocols, a simple text file, containing the
1414 necessary instructions to automate the filling and aspiration cycles to
1415 build experiments.

1416 Two versions of the software are available, one running on a mod-
1417 ern desktop computer that can drive four Dual, another running on a
1418 Raspberry Pi4 that drives only one Dual. The latest solution offers better
1419 scalability since each Dual is independent. A custom version of Ubuntu
1420 20.04 is preinstalled with the camera's SDK and the software. It can be
1421 downloaded at https://github.com/LJPZebra/dual_control/releases
1422 and flashed on an SD card or USB device. It is designed to work with a
1423 7-inch touchscreen display allowing a compact and easy-to-use control.

1424 **7.1.3 Construction and usage**

1425 For our needs, we have built four Duals that we ran in parallel. The
1426 construction requires a laser cutting machine, a 3D printer, and workshop

¹⁴²⁷ tools. It took about two weeks to build the four devices. The construction
¹⁴²⁸ does not require any specific knowledge, and access to a FabLab is suffi-
¹⁴²⁹ cient to carry out this project and find the required tools and help in the
¹⁴³⁰ event of difficulties.

¹⁴³¹ In practice, using Dual is effortless. Once the experiment template is
¹⁴³² created, the only manual task remaining is to place the fish in the milliflu-
¹⁴³³ idic chip, close the device, and play the experiment template. It is also
¹⁴³⁴ necessary to check that the suction containers neither run out of water
¹⁴³⁵ or chemical. It is then possible to do experiments with little as possible
¹⁴³⁶ dead times and minimal manual interventions.

¹⁴³⁷ A recurring problem we have encountered is the fouling of the mil-
¹⁴³⁸ lfluidic system. The dye used to visualize the flow ends up clogging the
¹⁴³⁹ valves and tubes and can stop the system in the middle of an experiment.
¹⁴⁴⁰ This problem can be solved by taking preventive habits. After each day of
¹⁴⁴¹ experiments, the millifluidic system has to be flushed with water. It can
¹⁴⁴² be performed automatically using an experimental template designed
¹⁴⁴³ to wash all the microvalves thoughtfully. Microvalves can regularly be
¹⁴⁴⁴ passed in an ultrasonic bath to clean them. The tubes have to be changed
¹⁴⁴⁵ when worn out, which can append after several weeks of intensive usage.
¹⁴⁴⁶ Another encountered problem was the syringe plunger wear. After several
¹⁴⁴⁷ months of usage, the plunger's latex cap loses water-tightness, and the
¹⁴⁴⁸ plunger has to be replaced, which can be done very quickly in less than
¹⁴⁴⁹ 2 minutes.

¹⁴⁵⁰ **7.2 The Tropical River**

¹⁴⁵¹ Studying chemical perception and chemically-driven navigation in a
¹⁴⁵² turbulent aquatic environment like the one fish encounters naturally re-
¹⁴⁵³ quires an experimental device capable of creating controlled flows and
¹⁴⁵⁴ chemical jets. We have created an experimental device capable of deliver-
¹⁴⁵⁵ ing a temperature-controlled laminar flow while recording the fish in both

1456 visible and infrared light. An injection nozzle is used to create turbulent
1457 or laminar jets within this laminar flow.

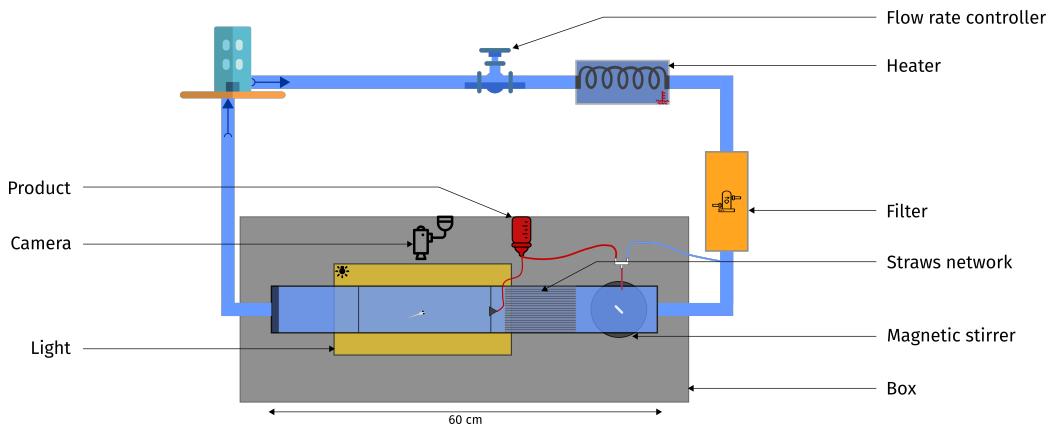


Figure 7.5: The Tropical River Schematic of the experimental setup.

1458 **7.2.1 Description**

1459 **Structure**

1460 The device's structure consists of a channel assembled from transparent
1461 laser cutted PMMA sheets fixed with Norcan rails to form a channel
1462 (60x10x10 cm), see Figure 7.5. A LED panel is placed under the channel to
1463 illuminate the setup by transmission in visible light. The setup can be illu-
1464 minated from above by a ring of infrared LEDs or by transmission by cov-
1465 ering the LED panel with a PMMA infrared transparent sheet. A Chameleon
1466 3 camera with a TODO lens is placed above the channel to record the ex-
1467 periments and connected to the computer via USB3. A mirror is placed
1468 at 45 degrees of inclination from the horizontal on the channel's side to
1469 control the fish's vertical position. The whole channel is set inside a box
1470 constructed with Norcan rails and plywood sheets to isolate the fish from
1471 the surrounding environment.

1472 **Hydrodynamic**

1473 The canal is supplied at one end with water from the building's water
1474 system. Before entering the canal, tap water is filtered by an activated
1475 carbon filter and heated by a water bath. A network of straws is placed in
1476 the channel to obtain a laminar flow, and a solenoid valve can adjust the
1477 flow rate. The other end of the channel is left free. The outgoing water
1478 is redirected to the building's wastewater network because the products
1479 tested do not require any special treatment before being disposed of. The
1480 water height inside the channel can be controlled by modulating the dyke
1481 height placed at the end of the channel.

1482 It is possible to dilute a product inside the channel by using an injec-
1483 tion nozzle located directly at the water supply outlet. A magnetic stirrer
1484 is placed inside the channel, before the straws network, to facilitate dilu-
1485 tion. Another injection nozzle can be placed inside the channel, after the
1486 straws network, to create a turbulent or laminar jet. It is supplied by an
1487 external tank, and the flow rate can be adjusted by gravity and delivery
1488 automatically controlled by a valve system.

1489 **Software**

1490 The control software allows to retrieve and control all the variables of
1491 the experiment. By default, it allows selecting the flow rate, temperature,
1492 injection valves, and camera settings. A vital function of the software is
1493 the ability to build experimental protocols. Easy to build, the protocol
1494 template is a simple text file specifying for a given device a variable's
1495 desired value at a given time point. Any Arduino sensor or control de-
1496 vice that follows a convention detailed in https://github.com/LJPZebra/the_tropical_river_control can be called in these protocols without
1497 the need to modify the software.
1498

1499 **Camera**

1500 The camera's options are directly accessible inside the software us-
1501 ing the Spinnaker SDK. Metadata like temperature, relative times to the
1502 experiment, flow rate, or user-specified value can be saved inside the im-
1503 age.

1504 **Temperature**

1505 The temperature regulation is made using a coiled heat exchanger
1506 tube added to the water inlet and immersed into a Neslab RTE water bath
1507 capable of cooling and warming. A temperature sensor is placed inside
1508 the channel and sends the instantaneous temperature to the control soft-
1509 ware. The software can control the water bath via an RS232 serial connec-
1510 tor selecting the temperature using a PID feedback system. Despite large
1511 variations of the building's water temperature, this system allows a pre-
1512 cise and rapid temperature regulation.

1513 **7.2.2 Usage and limitation**

1514 This experimental device is, in practice, very versatile. The ability to
1515 control the velocity and the temperature of the laminar flow, as well as
1516 the injection of products in an automated and quantified manner while
1517 recording the fish, allows the study of a wide variety of behaviors like
1518 rheotaxis, chemotaxis, and thermotaxis. The height of water in the channel
1519 can be modulated, making it possible to use adult and larval fish in the
1520 same setup.

1521 The insertion of chemical in the flow does not allow us to reuse the wa-
1522 ter. This is why the water supply is done using the water network of the
1523 building. Although filtered, the water quality depends on the building's
1524 water quality, which is usually not a problem for juvenile and adult fish

1525 that are reared in filtered tap water from 2 weeks onwards. However, this
1526 can be more problematic for larvae, which are more fragile and require
1527 calibrated water (E3). Although the experiment's duration is limited only
1528 by the computer's storage, small air bubbles appear in the channel after
1529 a few hours. This phenomenon is due to the dissolved gases present in
1530 the tap water and is detrimental to the fish. The activated carbon filter re-
1531 duces this phenomenon, but it remains present and limits the maximum
1532 experimental time to a few hours.

¹⁵³³ Chapter 8

¹⁵³⁴ Results

¹⁵³⁵ 8.1 Methods

¹⁵³⁶ In the next sections, we will present the experimental protocol that we
¹⁵³⁷ used and the analysis methods we developed to assess zebrafish chemi-
¹⁵³⁸ cal preference.

¹⁵³⁹ 8.1.1 Experiment

¹⁵⁴⁰ **Protocol** Fish chemical preference was assessed using Dual setup in a
¹⁵⁴¹ one-hour long experimental in a protocol detailed in Figure 8.1. This pro-
¹⁵⁴² tocol is subdivided into 4 cycles of 15 minutes each:

- ¹⁵⁴³ • B1: a cycle where buffer is injected on the two sides that serves as
¹⁵⁴⁴ a control.
- ¹⁵⁴⁵ • P1: a cycle where a product is injected on one side and water on the
¹⁵⁴⁶ other.

- 1547 • B2: same cycle as B1 that will serve to flush the system of any resid-
 1548 ual product.
- 1549 • P2: same as P1 but with sides inverted.

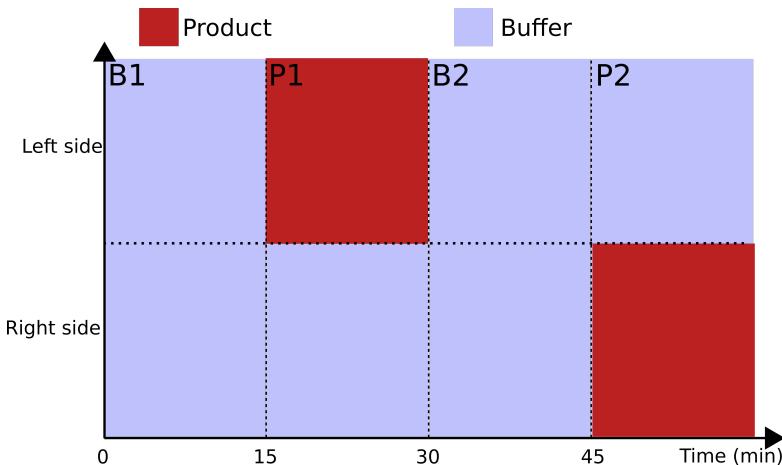


Figure 8.1: Protocol Experimental protocol used to assess zebrafish chemical preference. Product cycles P1 and P2 were regularly inverted to avoid any side bias.

1550 **Fish** In the following experiments, we use larval (6-8 days post-fertilization)
 1551 and early juveniles (14 to 21 weeks post-fertilization) zebrafish. Prefer-
 1552 ences were assessed without changes in rearing condition before the ex-
 1553 periment, except for ATP and adenosine, where fish were starved 24 hours
 1554 before the experiment, a protocol inspired from [50]. Chemicals were dis-
 1555 solved in a buffer the day of the experiment: E3 [] for larvae and degassed
 1556 tap water for juveniles.

1557 **Flow visualization** All the experiments were performed in the dark to
 1558 isolated chemical senses from other sensory inputs, in particular visual
 1559 perception. The flow was visualized using an infrared dye, an emulsion of
 1560 silicone oil, and infrared lightening. TODO(completer avec info Laeticia)

1561 **8.1.2 Analysis**

1562 The project's original idea was to use FastTrack ?? and an automatic
1563 custom-developed image analysis pipeline to monitor the fish and inter-
1564 face position. It would have made it possible to assess fish preference
1565 and decipher how the fish modulate its behavior when in contact with a
1566 chemical. However, in particular cases, the fish's complex behavior cou-
1567 pled with the challenge to get the interface position accurately pushed us
1568 to develop a manual method of analysis that we have used to characterize
1569 more precisely the fish behavior.

Time-base preference index The preference index is a measure of the attraction or repulsion of a product. It is defined from the times spent inside and outside the product as follows:

$$PI = \frac{t_{product} - t_{buffer}}{t_{product} + t_{buffer}}$$

1570 When $PI = 1$, the fish spend all its time on the product side; thus, the
1571 product is attractive. When $PI = -1$, it is the complete opposite, and the
1572 product is repulsive. $PI = 0$ means that the fish spent the same amount
1573 of time on each side; thus, the product is neutral.

1574 The critical point of this analysis method is to define when the fish
1575 is in the product. A coarse-grained analysis would be to assimilate the
1576 interface with its median position. Most of the time, it is accurate and
1577 reflects the fish preference well. On the other hand, it will provide no in-
1578 sight at the crucial moment where fish take a decision, i.e., when the fish
1579 cross the interface. We try unsuccessfully to develop an automatic inter-
1580 face extraction using the difference in contrast between buffer and water.
1581 But, complex behaviors of the fish at the interface that were challenging
1582 to quantify automatically, the large amount of data, and time constraints
1583 pushed us to use a manual method of quantification.

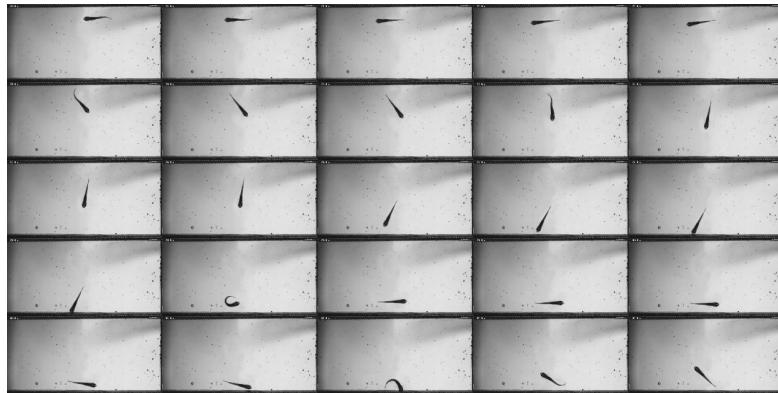


Figure 8.2: Complexe Example of a complex behavior where the fish swim inside the interface. The fish is creating advection and automatically extract the interface position is challenging.

1584 **Event analysis** To be time-efficient, we choose to develop a manual anal-
 1585 ysis method. We choose 4 characteristics and meaningful events, see Fig-
 1586 ure 8.3, to quantify the behavior: when the fish cross the interface to
 1587 change side (PB, BP) and when the fish sense the interface and return
 1588 to the side where it was (BB, PP). By recording these events, we can quan-
 1589 tify the moment where the fish make a decision, which was not possible
 1590 with the time-based analysis.

1591 To count these events, we performed a blind analysis where movies
 1592 were anonymized and scrambled. The events counting was performed by
 1593 two independents individuals and then pulled together to get the average
 1594 and deanonymized.

1595 We look first at the correlation between the two independents analy-
 1596 ses, see Figure 8.4, to check for human bias. The correlation between each
 1597 fish's events count was equal to 0.97, indicating that there is no significant
 1598 bias or disagreement between the analysis.

We can define an event-based preference index as follow:

$$PI = \frac{n_{BP} + n_{PP} - n_{BB} - n_{PB}}{n_{BP} + n_{PP} + n_{BB} + n_{PB}}$$

1599 This preference index is slightly different from the time-based one. It will

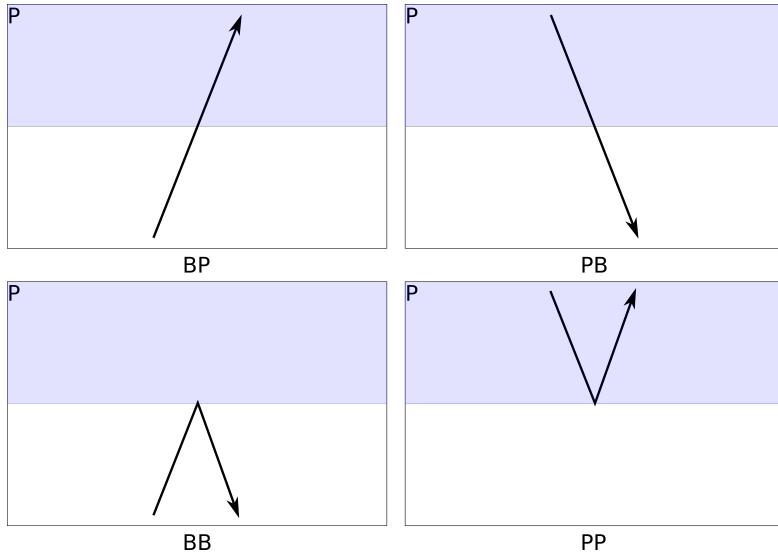


Figure 8.3: Events The event-based method of quantification distinguish 4 types of events that can occurs during P1 and P2 product cycles.

1600 only consider fish decisions and not be impacted by the fish inactivity
 1601 and freezing.

1602 **Ratio exploration-exploitation** Besides the preference of the fish, an
 1603 interesting quantity that we can look for is the type of behavior. We can
 1604 distinguish two modes of behavior: an exploration mode where the fish
 1605 cross the interface to explore the environment and an exploitation mode
 1606 where the fish stay on the same side and make touch-and-turn at the
 1607 interface. Note that this quantity is by any means a preference indicator.

The ratio exploration-exploitation can be computed based on the event as follows:

$$r = \frac{n_{BP} + n_{PB}}{n_{BB} + n_{PP}}$$

1608 When $r = 1$ there is as much exploration as exploitation, $r < 1$ there is
 1609 more exploitation and $r > 1$ more exploration. In the event where $n_{BB} +$
 1610 $n_{PP} = 0$ we take $r = 0$ (pure exploration), and $n_{BP} + n_{PB} = 0$ we take $r = \infty$
 1611 (pure exploitation).

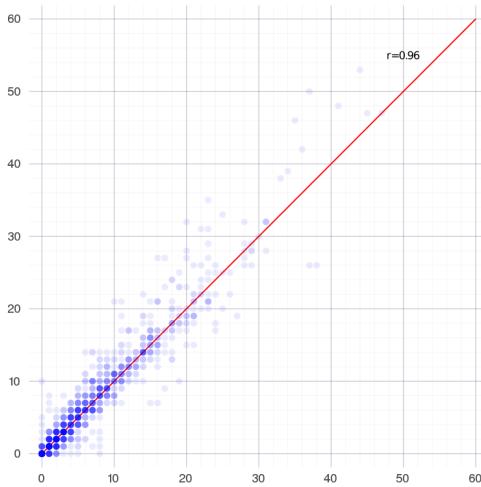


Figure 8.4: Analysis correlation Correlation between the two independent event-based analysis. One point is the count for one event type for one fish. Red line is the identity line. The point color intensity is encoding the number of points at the same coordinate.

Model Based on the events recordings, we can build a two-state Markov chain model (Figure 8.5). We can define the probability of transfer as follows:

$$p = \frac{n_{BP}}{n_{BP} + n_{BB}}$$

$$b = \frac{n_{PB}}{n_{PB} + n_{PP}}$$

¹⁶¹² With p the probability of going from the buffer to the product and b the
¹⁶¹³ probability of going from the product to the buffer.

¹⁶¹⁴ With this model, the preference index can be defined directly by $p - b$
¹⁶¹⁵ the proportion of time spent in the product state minus the proportion
¹⁶¹⁶ of time spent in the buffer state. If p (resp. b) can be defined, we take
¹⁶¹⁷ $PI = -1$ (resp. $PI = 1$).

An indicator of the fish exploration-exploitation behavior can be derived as follows:

$$r_{markov} = 2\text{Min}(p, b) - 1$$

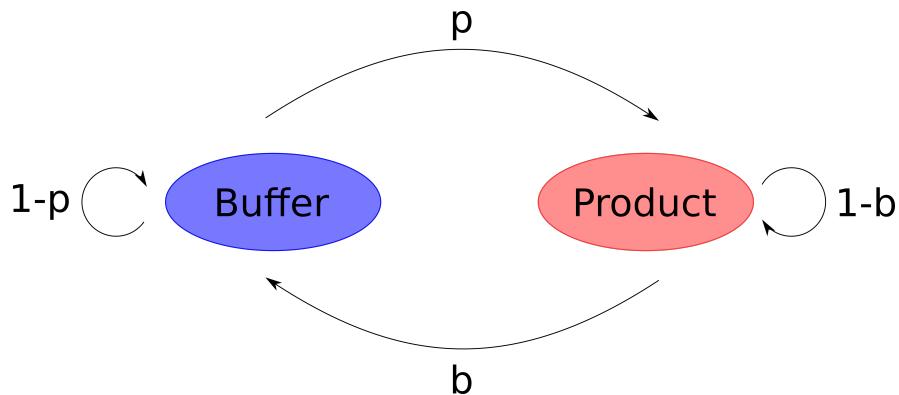


Figure 8.5: Two states Markov chain model

We can build a numerical simulation of two states Markov chain to explore the relationship between p and b probabilities, the preference index, and r_{markov} . The numerical simulation will construct Markov chain of probabilities p , and b . The length of the chain is drawn from the experimental chain length distribution that follows a discrete negative binomial distribution (see Figure 8.6):

$$f(k) = \binom{k+n-1}{n-1} a^n (1-a)^k$$

1618 With $n = 2.527$ et $a = 0.084$.

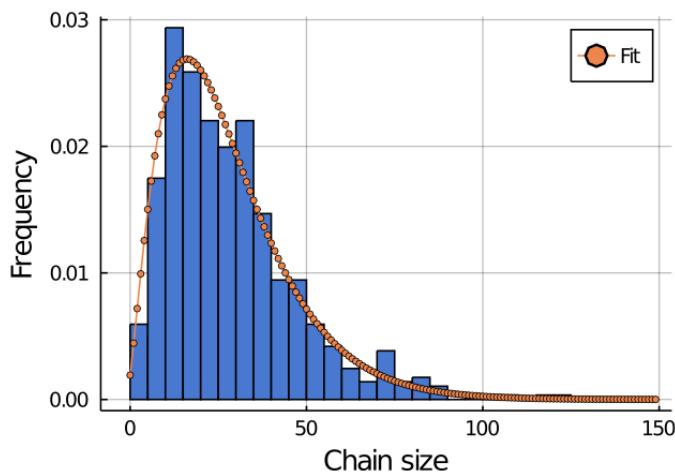


Figure 8.6: Chain length Chains length distribution.

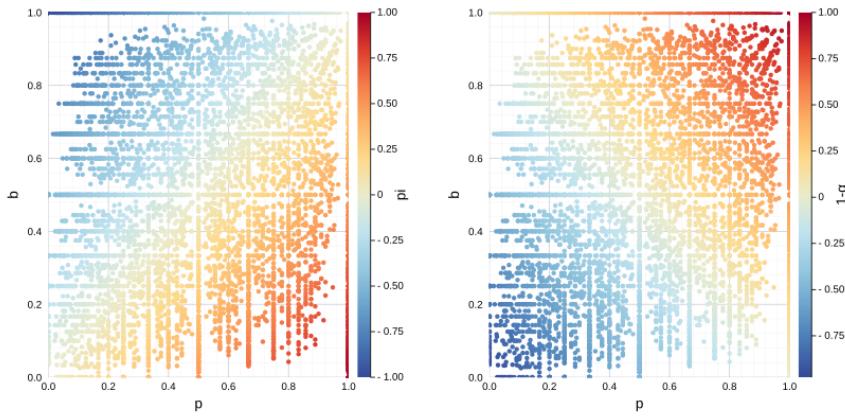


Figure 8.7: Numerical simulation Preference index and r_{markov} for 10 000 realisations with a chain size following the experimental distribution.

1619 We see on Figure 8.7 that there are forbidden couples of (p, b) due to
 1620 the chain length. The preference indexes are distributed on either side of
 1621 the identity line $p = b$ where $p_i = 0$ with an upper triangle repulsive and
 1622 a lower triangle attractive. Looking at the map of r_{markov} , we see a right
 1623 upper corner at high p and b dominated by exploration, and a bottom left
 1624 corner by exploitation. As expected, the fish can have a neutral preference
 1625 with either exploration or exploitation, but a strong preference can only
 1626 happen in a regime dominated by exploitation.

1627 8.2 Results

1628 8.2.1 Setup characterisation

1629 **Left-right bias** We started by checking the preference index in the B1
 1630 buffer cycle to capture an eventual left-right bias of setup. There is water
 1631 on the two sides in this control cycle, and fish were never exposed to
 1632 any product. We compute the time-based preference index choosing the
 1633 middle of the tank as left-right separation (in B1, there is no dye, thus no

1634 real separation). Figure 8.8 presents the distribution of preference index
 1635 for B1 cycles for larvae and juveniles. The first thing we can notice is the
 1636 significant variability among fish. The median value ($pi = 0.08$ for $N = 125$
 1637 for larvae and $pi = 0.05$ for $N = 178$ juveniles) is close to zero that will
 1638 exclude any systematic left-right bias of the setup.

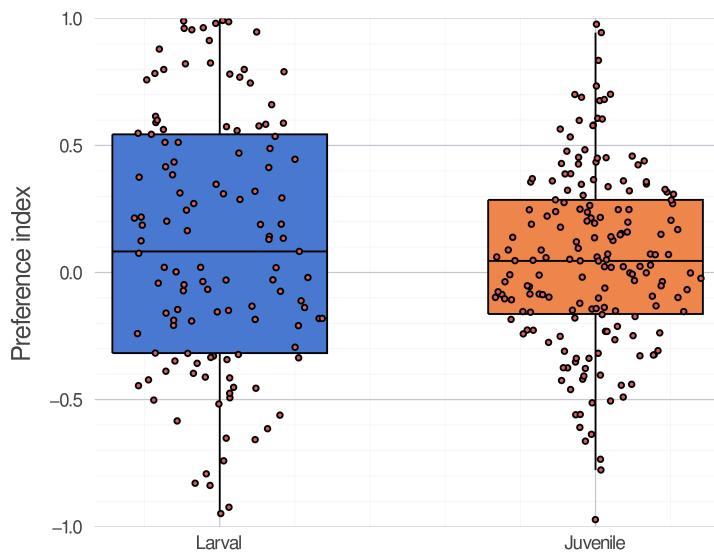


Figure 8.8: Left-right bias Distribution of the time preference index computed for the first buffer cycle (B1) with water on the two sides. Each point is a value of preference index for one fish. Black line is the distribution median.

1639 **Dye neutrality** In the product cycles P1 and P2, the dye is diluted with
 1640 the product to visualize the flow. To study the dye's impact on the fish, we
 1641 performed control experiments with juvenile fish assessing only the dye
 1642 with the protocol presented above.

1643 Figure 8.9A presents the event-based preference index for the dye only.
 1644 Each point is a fish for one independent manual analysis; the point's
 1645 size encodes the number of events during the cycle. We see that the
 1646 dye seems neutral to the fish with no clear attraction or repulsion. As
 1647 expected, there is large inter-fish variability. The preference index distri-
 1648 butions for P1 and P2 are not statistically significant, but cycle P2 includes

₁₆₄₉ more outliers fish with strong preferences.

₁₆₅₀ Figure 8.9B presents the time-based preference index, each point rep-
₁₆₅₁ resenting one fish. We see a slight repulsion in the P1 cycle and a slight
₁₆₅₂ attraction in the P2 cycle toward the dye; there is no clear, robust pref-
₁₆₅₃ erence toward the dye. These slight preferences are not visible with the
₁₆₅₄ event-based analysis and come from the time-based method that is less
₁₆₅₅ precise with our low statistics $N = 13$ and more sensible to periods where
₁₆₅₆ the fish do not make a decision.

₁₆₅₇ Figure 8.9C presents the p and b probabilities from the Markov chain
₁₆₅₈ model by fish. In each cycle, we see that $p \approx b \approx 0.65$, meaning that the
₁₆₅₉ fish can perceive the dye ($p = b = 1$ mean that the fish do not perceive the
₁₆₆₀ dye) or at least the side change by another unknown mean. This effect is
₁₆₆₁ also visible on the ratio exploration-exploitation (Figure 8.9D).

₁₆₆₂ Altogether, these results indicate that the fish can sense the dye but
₁₆₆₃ without manifesting any preference. With enough statistics to mitigate the
₁₆₆₄ fish inter-variability, we will be able to assess the fish preference while
₁₆₆₅ visualizing precisely the flow.

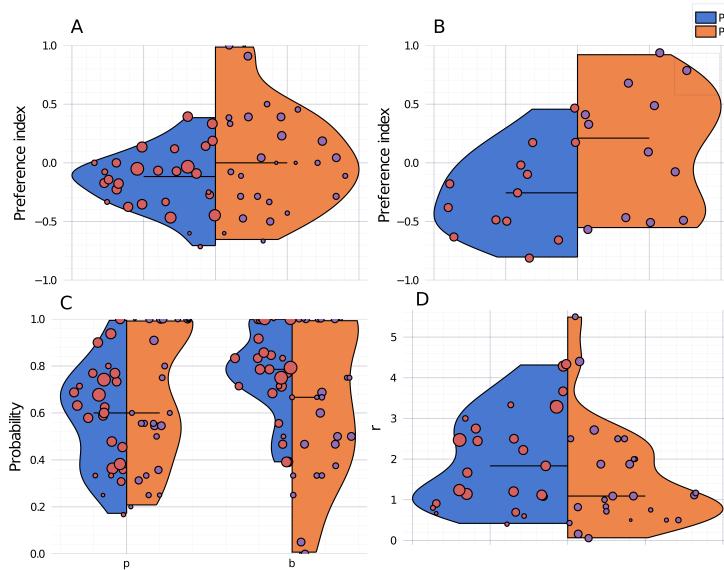


Figure 8.9: Dye bias **A.** Event-based preference index, one point is representing one fish for one independant analysis. **B.** Time-based preference index, one point is one fish. **C.** p and b probabilities from the Markov chain model, one point is representing one fish for one independant analysis. **D.** Ratio exploration-exploitation, one point is representing one fish for one independant analysis.

1666 8.2.2 Products screening

1667 We have screened several products known to elicit an aversion or at-
 1668 traction to the zebrafish. Because the literature is mainly focused on adult
 1669 fish, our strategy was to assess early juvenile fish preference first and then
 1670 reduce fish age.

1671 **Citric acid juvenile** Citric acid is known to be repulsive in many fish
 1672 species. It was shown that zebrafish could sense their environment pH
 1673 [112, 113] and display a strong aversion for acidic pH ($pH \approx 3$). As a pos-
 1674 tive control, early juveniles and larval zebrafish preference were assessed
 1675 using citric acid solutions ranging from $5 \times 10^{-3} M$ to $1 \times 10^{-6} M$, ie pH from
 1676 2.8 to 4.2 (Figure 8.10).

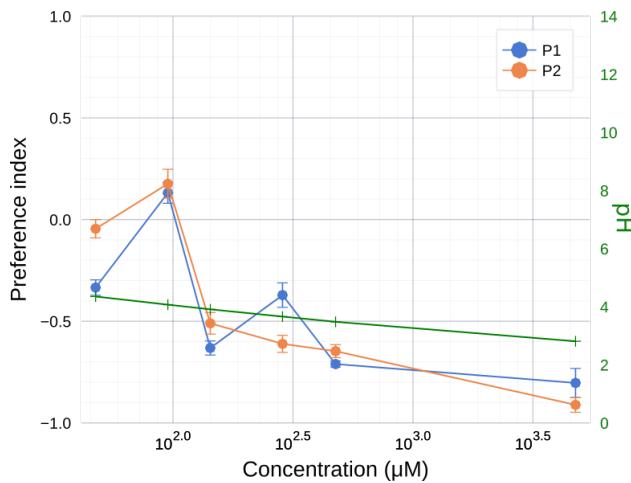


Figure 8.10: Citric acid: time-based preference index Time based preference index for P1 and P2 cycle (mean \pm SEM), pH in green.

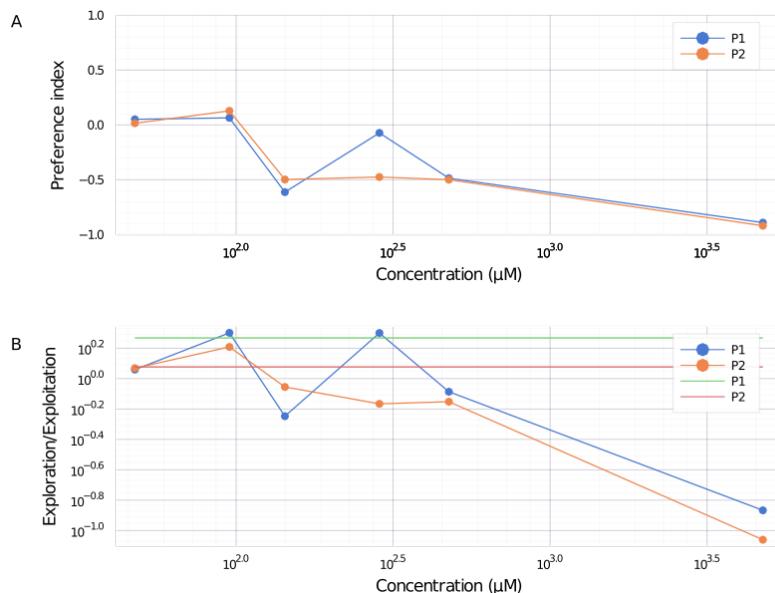


Figure 8.11: Citric acid: event-based analysis A. Mean event-based preference index. B. Mean ratio exploration-exploitation.

1677 The ratio exploration-exploitation (Figures 8.11B F.1B) showed an ap-
 1678 parent decrease when the citric acid concentration increase. Fish can
 1679 sense citric acid and reduce their exploration to the benefit of a stereo-
 1680 typed exploitation behavior where they often go to the interface and re-

1681 turn in the buffer (see Appendix ?? for a detailed description).

1682 The time-based (Figure 8.10A), event-based (Figure F.1B) and model-
1683 based (Figure 8.10A) mean preference indexes decrease in a concentration-
1684 dependent manner. The distributions of preference indexes by fish (Fig-
1685 ure G.1) show a decrease in variability as the concentration increase and
1686 no significant difference between P1 and P2 cycles, which tends to sug-
1687 gest that fish adopt the same characteristic behavior with less variability
1688 as a preference emerged.

1689 **Citric acid larvae** Preliminary experiments were done to assess the ef-
1690 fect of citric acid on zebrafish larvae (Figure G.1). Despite low statistics
1691 (see section 9) and significant variability, we see that citric acid seems to
1692 be repulsive as expected from juvenile results. More experiments need to
1693 be done to confirm this effect and account for the larvae inter-variability.

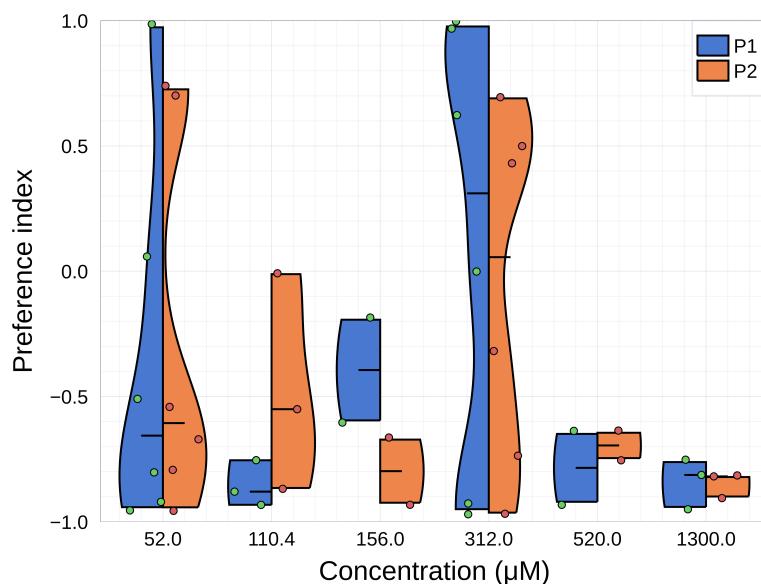


Figure 8.12: Citric acid: preference index for larvae Time-based preference index, one point is one fish.

1694

1695 Using Dual, we were able to assess the repulsion to citric acid by ze-
1696 brafish. Our analysis and model were able to capture the fish behavior
1697 changes and quantify the repulsion magnitude. As expected, the inter-
1698 fish variability is large but seems to be reduced by aversion where fish
1699 adopt a stereotyped exploitation behavior.

1700 **ATP and adenosine** Zebrafish have pear-shaped olfactory sensory neu-
1701 rons expressing olfactory receptors specific to adenosine [50]. ATP, ADP,
1702 and AMP are dephosphorylated in the zebrafish olfactory epithelium by
1703 two kinases [50]. Behavioral responses to adenosine and ATP were shown
1704 to exist with arousal and food-seeking behavior. We assessed preference
1705 for ATP and adenosine for larval and early juveniles zebrafish. Fish were
1706 starved 24 hours before the experiment, and preference was assessed us-
1707 ing the protocol presented below.

1708 **Adenosine juveniles** Fish can perceive adenosine as indicated by the
1709 ratio exploration-exploitation (Figures 8.14B F.2B) that decrease with the
1710 concentration and are lower than the control (only dye).

1711 The event-based and model-based preference indexes (Figures 8.14A F.2A)
1712 show a slight repulsion for the P1 cycle and a slight attraction for the P2
1713 cycle at the two lower concentrations. The time-based preference index
1714 (Figures 8.13) displays a slight repulsive preference for the P1 cycle. There
1715 is no clear preference, and analysis from event-based and time-based
1716 preferences disagree, meaning that there is probably not enough statis-
1717 tic to conclude. Besides, preference index distributions (Figure G.2) show
1718 great variability. As no clear preference emerged from the early experi-
1719 ments, we focused on the concentration $50\mu M$ and increased the statis-
1720 tics up to $N = 28$. We see a slight repulsion on the P1 cycle at the same
1721 level that the other concentration. However, the P2 cycle shows a clear
1722 repulsion with a median preference index equal to -0.6 and 2 clusters
1723 of fish, one attracted and one repulsed. Unexpectedly, adenosine does
1724 not seem to attract early juvenile zebrafish. On the contrary, at high con-

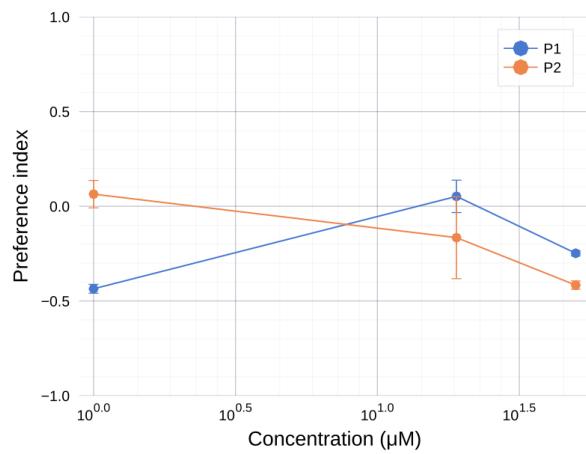


Figure 8.13: Adenosine: time-based preference index. Time based preference index for P1 and P2 cycle (mean \pm SEM).

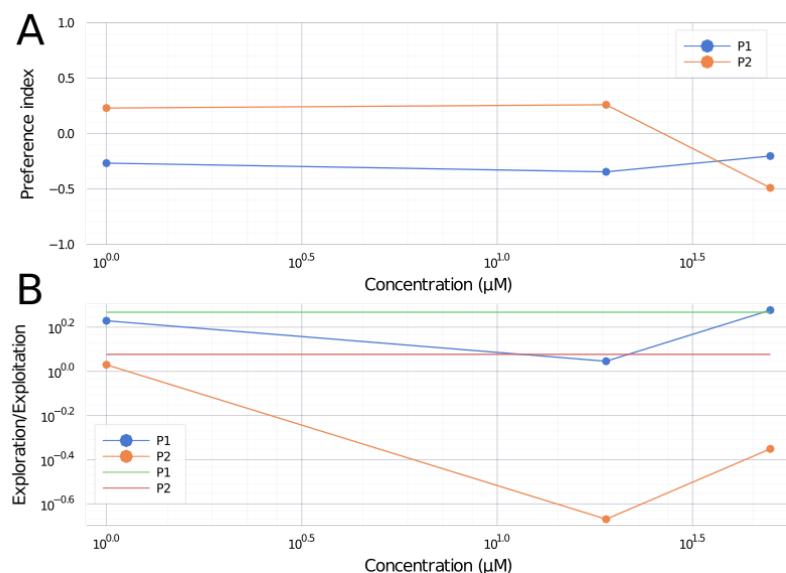
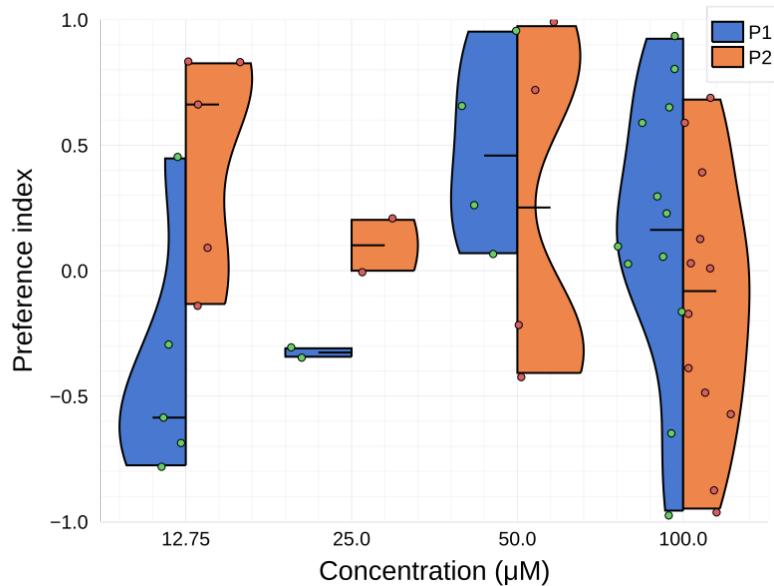


Figure 8.14: Adenosine: event-based analysis A. Mean event-based preference index. B. Mean ratio exploration-exploitation.

1725 concentration, it is clearly repulsive. However, one part of the fish ($\approx 30\%$)
 1726 seems to be either attracted or neutral at the second presentation of the
 1727 product.

1728 **Adenosine larvae** Zebrafish larvae were exposed to adenosine with the
1729 same protocol as juveniles. Preliminary results (Figure 8.15) based on the
1730 time preference index seems to indicate no strong preference at any con-
centration.



1731 **Figure 8.15: Adenosine: preference index for larvae** Time-based preference index.
One point is one fish, black line is the distribution median.

1732 **ATP juveniles** Interestingly, we find that ATP, a product that was shown
1733 to be attractive on adults, produces a two phases behavior on early ju-
1734 venile zebrafish (Figures 8.16A 8.17A F.3A). At low concentration, fish do
1735 not manifest a preference, but at the highest concentration ($125\mu\text{M}$ and
1736 $200\mu\text{M}$), the P1 cycle became repulsive and the P2 cycle attractive. Com-
1737 paring the distribution of event-based and time-based preference index
1738 by fish (Figure G.3), we see that this effect is significant at $125\mu\text{M}$ and
1739 $200\mu\text{M}$ (one-sided Wilcoxon signed-rank test). A fish by fish analysis (Fig-
1740 ure 8.18A) reveals that the fish proportion that changes its preference to-
1741 ward attraction increase with the concentration. (Figure 8.18B) represents
1742 the p and b probability from the markov chain model. We can see that a

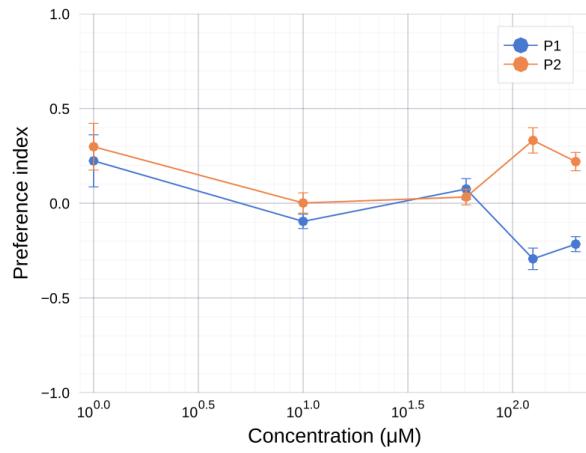


Figure 8.16: ATP: time-based preference index.

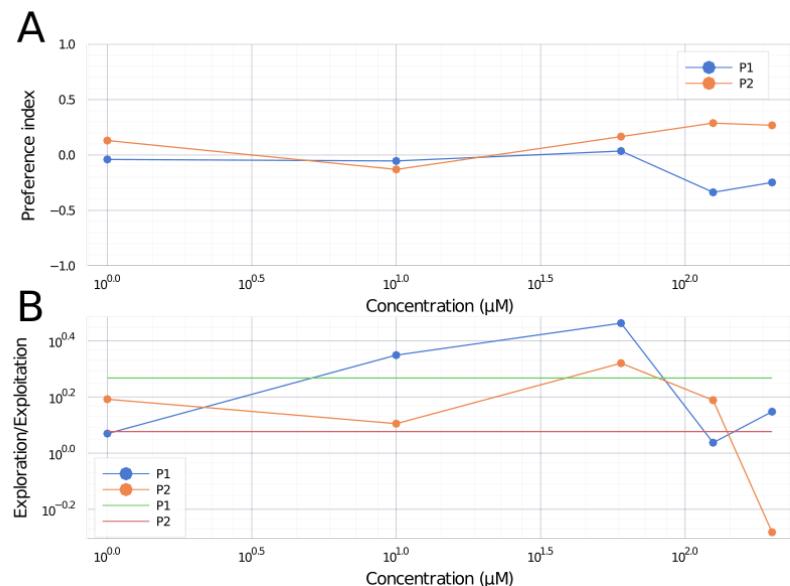


Figure 8.17: ATP: event-based analysis A. Mean event-based preference index. B. Mean ratio exploration-exploitation.

1743 lot of fish cross the identity line between the P1 and P2 cycle to invert
1744 their preference indexes.

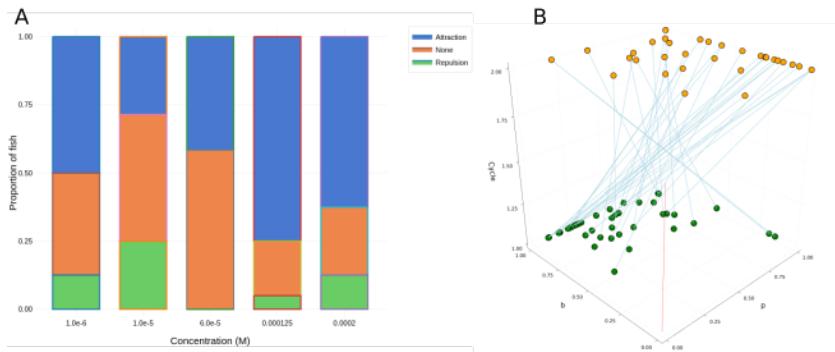


Figure 8.18: Cycle effect **A.** Fish proportion changing preference between the P1 and P2 cycles. **B.** Probabilities p and b for 125 and $200\mu M$, one point is one fish for one independent analysis ($2N$ fish).

1745 **ATP larvae** Zebrafish larvae were exposed to ATP with the same protocol
 1746 as juveniles. Preliminary results (Figure ??) based on the time preference
 1747 index indicate a repulsion at every concentrations. High concentrations
 1748 seem to produce more robust repulsive behavior with low variability.

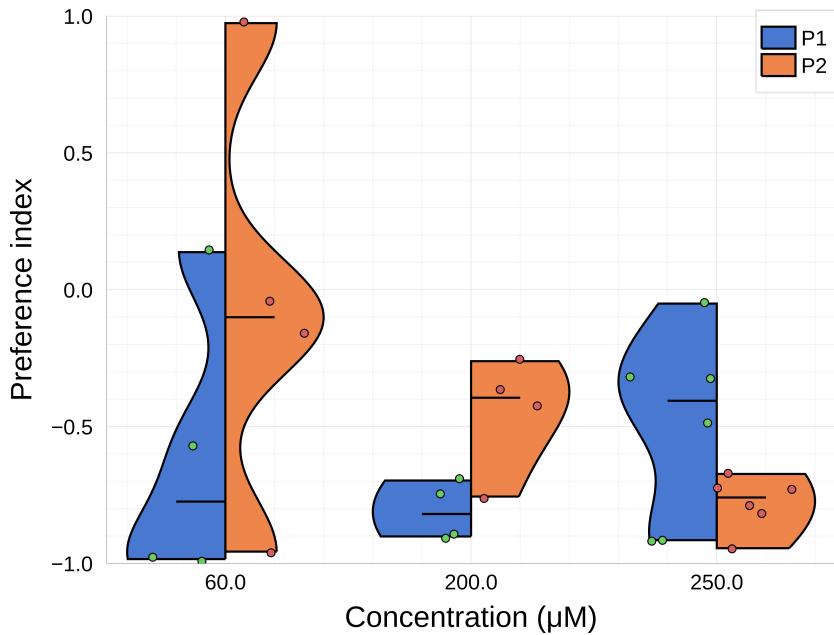
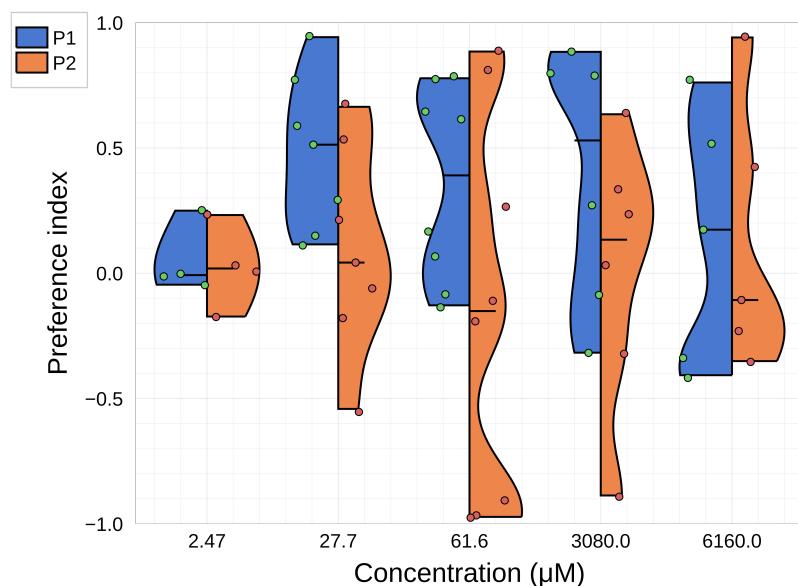


Figure 8.19: ATP: preference index for larvae Time-based preference index.

1749 Not documented in the literature to our knowledge, an interesting ef-

1750 fect was found on juvenile zebrafish exposed to ATP with an inversion of
1751 preference at the second product presentation. Experiments with larvae
1752 do not seem to indicate the same effect. More experiments need to be
1753 done, especially on larvae and at intermediate ages, to better characterize
1754 this behavior.

1755 **Quinine juveniles** Quinine is a natural alkaloid that tastes bitter, and
1756 that is a highly deterrent substance for many fish species [115]. Unexpect-
1757 edly, we see with the time-based preference index (Figure 8.20) that the
1758 preference seems to be mostly neutral. A slight attraction can be con-
1759 stated on the P1 cycle, and a neutral a slightly repulsive preference on the
1760 P2 cycle. Despite relatively good statistics (≈ 7 fish by concentration), no
1761 robust preference emerged as we see with a large variability in responses.



1762 **Figure 8.20: Quinine: preference index for juveniles** Time-based preference in-
dex. One point is one fish.

Chapter 9

Conclusion

¹⁷⁶³ In these chapters, we saw how we built two experimental setups to assess and understand the chemical preference of larval and early-juveniles zebrafish.

¹⁷⁶⁸ The Tropical River is a setup capable of producing controlled flows. ¹⁷⁶⁹ The flow temperature and velocity can be precisely controlled. Turbulent ¹⁷⁷⁰ and laminar jets can be created inside the flow to mimic more realistic ¹⁷⁷¹ fragmented chemical perception. This setup was originally developed to ¹⁷⁷² study the chemically-driven navigation after finding a robust, attractive ¹⁷⁷³ compound with Dual. Due to the difficulty of finding such a product, we ¹⁷⁷⁴ could not use it in the context of this project. Nonetheless, several preliminary ¹⁷⁷⁵ experiments were performed to check the setup, and it was used in ¹⁷⁷⁶ another project to study the fish swimming taking advantage of the flow ¹⁷⁷⁷ velocity control and high framerate imaging.

¹⁷⁷⁸ Dual is a high-throughput chemical screening setup that allows exploring ¹⁷⁷⁹ the combination of products, concentrations, and fish ages easily ¹⁷⁸⁰ and rapidly. The setup is scalable and can be built at the laboratory for ¹⁷⁸¹ less than 1500 USD. In the first part of the thesis, we built the setup and ¹⁷⁸² controlled that it did not have any bias. We check that the infrared dye ¹⁷⁸³ used to visualize the flow was neutral to the fish. We successfully use Dual

1784 to assess the chemical preference of several products. We show a strong
1785 aversion to citric acid and an exciting effect with ATP where fish inverts
1786 their preferences at the second product presentation. Work remains to be
1787 done to precisely characterize this effect. An interesting question would
1788 be to check what happened at the third presentation of ATP. It can be
1789 done easily by adding a cycle to the experiment. If this attractive prefer-
1790 ence persists in time, ATP could be used in The Tropical River to study
1791 chemically-driven navigation. A temporal characterization would also be
1792 of great interest. Adults were shown to be attracted by ATP without men-
1793 tion of prior exposures. Studying the effect at a fixed concentration, like
1794 $125\mu M$, by varying the fish age will help understand the development of
1795 ATP's perception from larvae to adults.

1796 Preliminary results shown with larvae need more statistics. Studying
1797 larvae poses several challenges. First, larvae tend to explore less than ju-
1798 veniles, and combined with their small sizes, they do not frequently cross
1799 the interface. We circumvented this problem by putting up to 4 larvae by
1800 experiment, thus increasing the statistics, but this adds work in the anal-
1801 ysis phase. Secondly, larvae are more susceptible to fatigue and freezing
1802 than juveniles. It frequently happens that larvae stop swimming and do
1803 not recover until the end of the experiment. For this reason, assessing the
1804 ATP effect on larvae is very challenging: reducing cycles length to reduce
1805 fatigue will lead to fewer crossing events, but longer cycles will lead to
1806 more fish dropouts before the cycle P2. For this reason, studying larvae
1807 will need many statistics, a thing that we had not the time to do in the
1808 context of this thesis.

1809 The analysis methods are fragmented according to fish ages and prod-
1810 ucts. We first implemented a coarse-grained approach where we recorded
1811 the fish's position and computed the time-based preference index tak-
1812 ing the mean interface position. Then we tried to refine this approach
1813 to better characterize the fish behavior at the interface by extracting the
1814 interface position. This was not practically achievable due to several con-
1815 straints: time, fish complex behavior, and challenging image processing

1816 problems on a large amount of data (8 To of data). Finally, we chose to
1817 perform a manual analysis that focuses on events happening at the in-
1818 terface, moments where the fish had to make a decision. It was possible
1819 to confirm the time-based analysis results with this approach and char-
1820 acterize the fish behavior more precisely. This approach was only applied
1821 to products where there were enough statistics and already interesting
1822 effects to delve into.

1823 Studying the chemical perception of zebrafish proved to be challeng-
1824 ing. The fish inter-variability is high, as we have seen by screening prod-
1825 ucts. In these experiments, we use 4 Duals to record up to 350 hours
1826 of usable experiments, but we see that more statistics are needed. We
1827 hope that Dual as a low-cost, easily replicable setup will encourage peo-
1828 ple to build and use the setup to better characterize zebrafish chemo-
1829 preference.

Appendices

¹⁸³¹ **Appendix D**

¹⁸³² **Visualisation**

¹⁸³³ **D.1 Boxplot**

¹⁸³⁴ The boxplot is a non-parametric method to represent data graphically
¹⁸³⁵ using their quartiles, its read as indicated in the Figure ^{D.1}.

¹⁸³⁶ **D.2 Violinplot**

¹⁸³⁷ The violinplot is a non-parametric method to represent data graph-
¹⁸³⁸ ically using their kernel density plot, see Figure ^{D.2}. The kernel density
¹⁸³⁹ estimation is a non-parametric method to estimate the probability den-
¹⁸⁴⁰ sity function. The width kernel density plot will give the frequency of a
¹⁸⁴¹ given value.

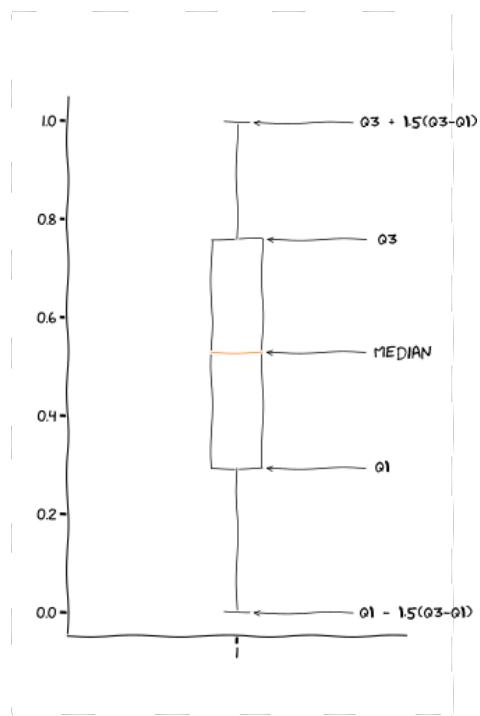


Figure D.1: Boxplot

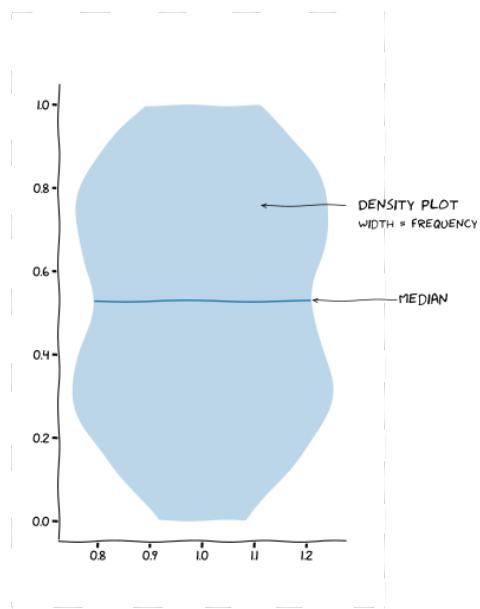


Figure D.2: Violinplot

¹⁸⁴² **Appendix E**

¹⁸⁴³ **Calculation**

¹⁸⁴⁴ **E.1 Mean quantities**

Mean quantities were calculated from events count following these definition:

$$\begin{aligned}\bar{P}\bar{I} &= \frac{\sum_{fish}(n_{BP} + n_{PP} - n_{BB} - n_{PB})}{\sum_{fish}(n_{BP} + n_{PP} + n_{BB} + n_{PB})} \\ \bar{r} &= \frac{\sum_{fish}(n_{BP} + n_{PB})}{\sum_{fish}(n_{BB} + n_{PP})} \\ \bar{p} &= \frac{\sum_{fish} n_{BP}}{\sum_{fish}(n_{BP} + n_{BB})} \\ \bar{b} &= \frac{\sum_{fish} n_{PB}}{\sum_{fish}(n_{PB} + n_{PP})}\end{aligned}$$

¹⁸⁴⁵ **E.2 Statistical tests**

¹⁸⁴⁶ Statistical significance was evaluated using the Wilcoxon signed-rank test, one-sided when knowing the effect's direction, or two-sided otherwise. In the case of the event-based analysis, the test was carried out

¹⁸⁴⁹ separately on the two independent manual counting then averaged to
¹⁸⁵⁰ avoid a size effect bias.

¹⁸⁵¹ E.3 Image analysis

¹⁸⁵² Fish head position was extracted using FastTrack. Each recording was
¹⁸⁵³ manually checked to remove low-quality movies either with imperfect
¹⁸⁵⁴ flows or non-responsive fish.

¹⁸⁵⁵ A custom image processing procedure was developed to extract the in-
¹⁸⁵⁶ terface using the contrast difference due to the dye see Figure ?? . First, the
¹⁸⁵⁷ maximal and minimal z-projection of the movie was calculated. For the
¹⁸⁵⁸ minimal projection, the fish is masked with a white mask on each image
¹⁸⁵⁹ before the projection, only keeping the flow and not projection the fish.
¹⁸⁶⁰ Each image of the movie is then normalized with minimal and maximal
¹⁸⁶¹ projection. Finally, the interface was detected using an Otsu threshold.

¹⁸⁶² A python command-line interface ¹ was created to export the relevant
¹⁸⁶³ data in a synthetic toml file easy exportable, one experiment by file. Sta-
¹⁸⁶⁴ tistical analysis and numerical simulations were performed using Python
¹⁸⁶⁵ and Julia programming language.

¹https://github.com/LJPZebra/dual_analysis

¹⁸⁶⁶ **Appendix F**

¹⁸⁶⁷ **Markov Model**

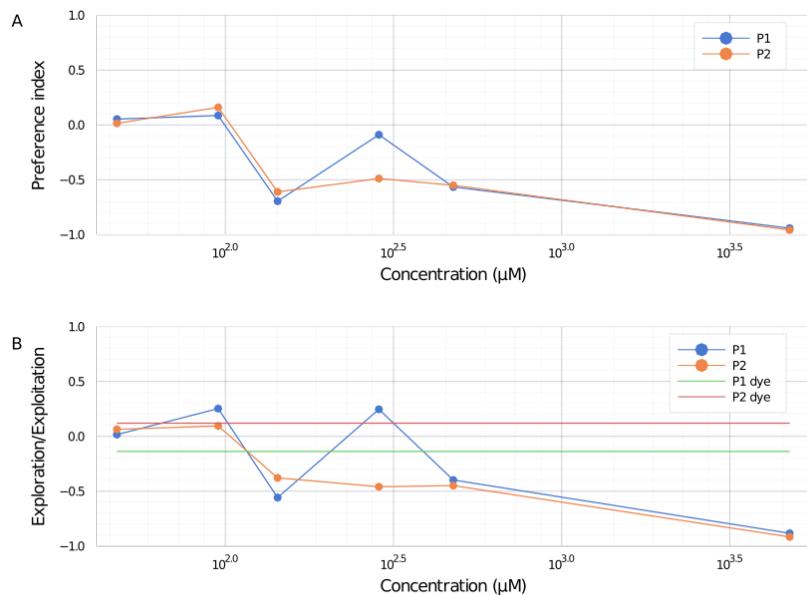


Figure F.1: Citric acid: Markov chain analysis **A.** Mean Markov chain preference index. **B.** Mean Markov chain ratio exploration-exploitation.

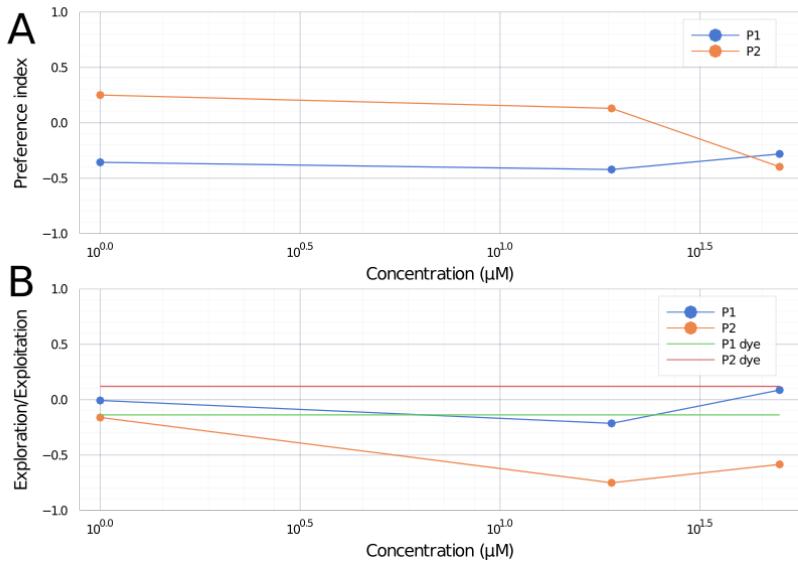


Figure F.2: Adenosine: Markov chain analysis **A.** Mean Markov chain preference index. **B.** Mean Markov chain ratio exploration-exploitation.

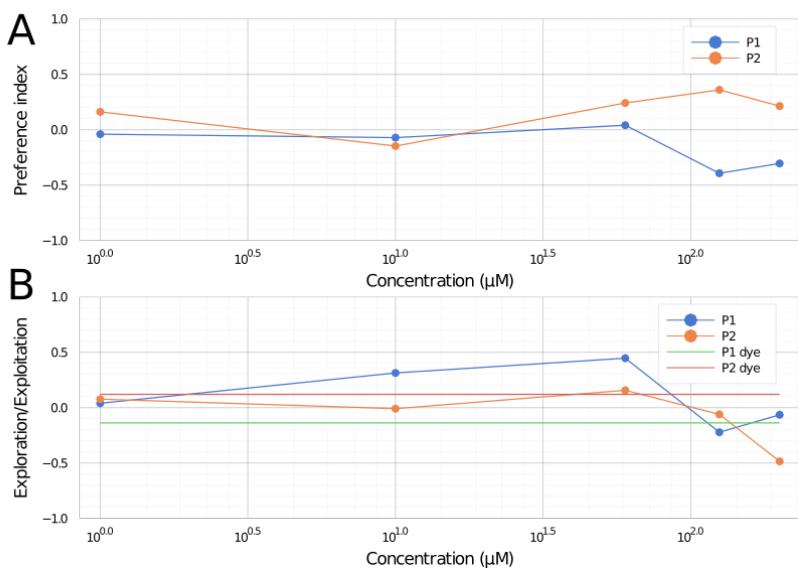


Figure F.3: ATP: Markov chain analysis **A.** Mean Markov chain preference index. **B.** Mean Markov chain ratio exploration-exploitation.

¹⁸⁶⁸ **Appendix G**

¹⁸⁶⁹ **Preference index distributions**

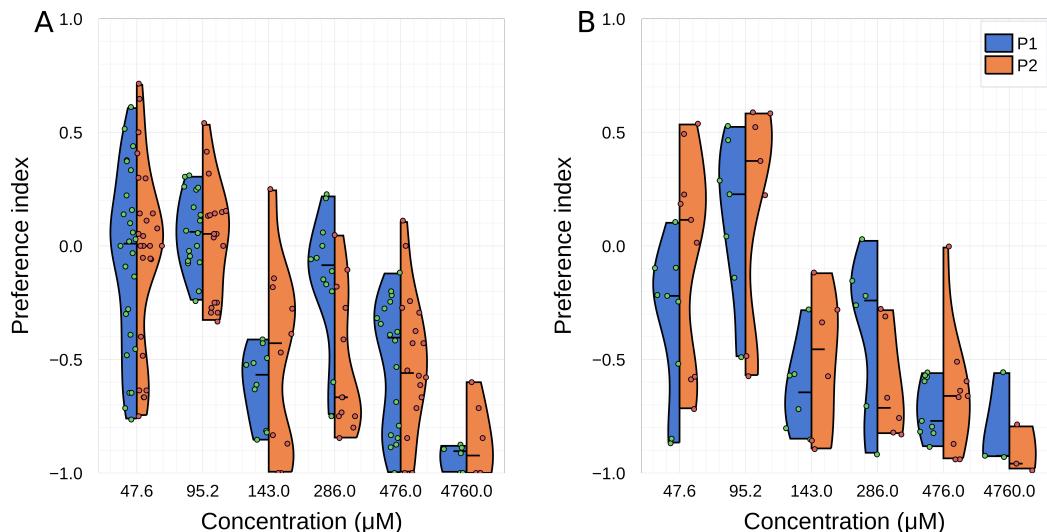


Figure G.1: Citric acid: preference index for juveniles **A.** Event-based preference index, one point is one fish for one independent analysis (total of $2N$ points). **B.** Time-based preference index, one point is one fish.

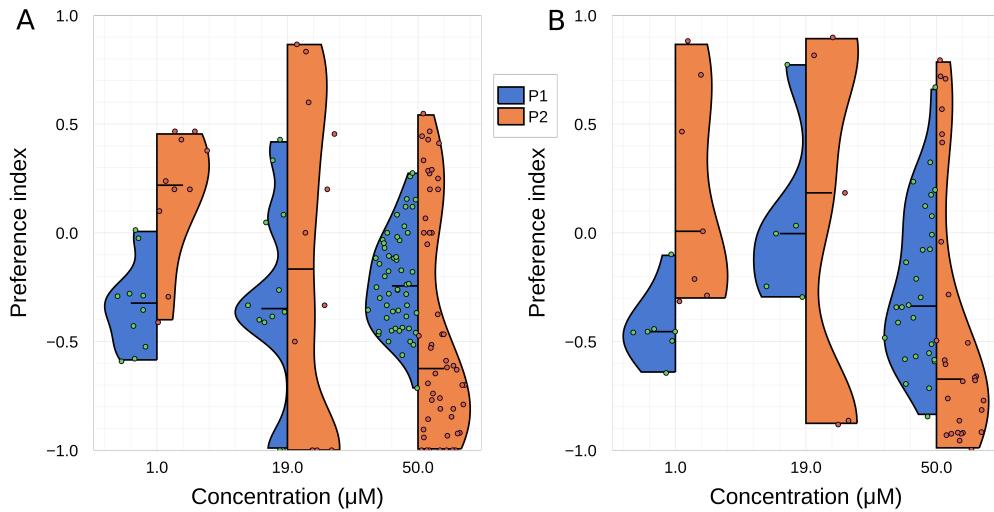


Figure G.2: Adenosine: preference index for juveniles **A.** Event-based preference index, one point is one fish for one independent analysis (total of $2N$ points). **B.** Time-based preference index, one point is one fish.

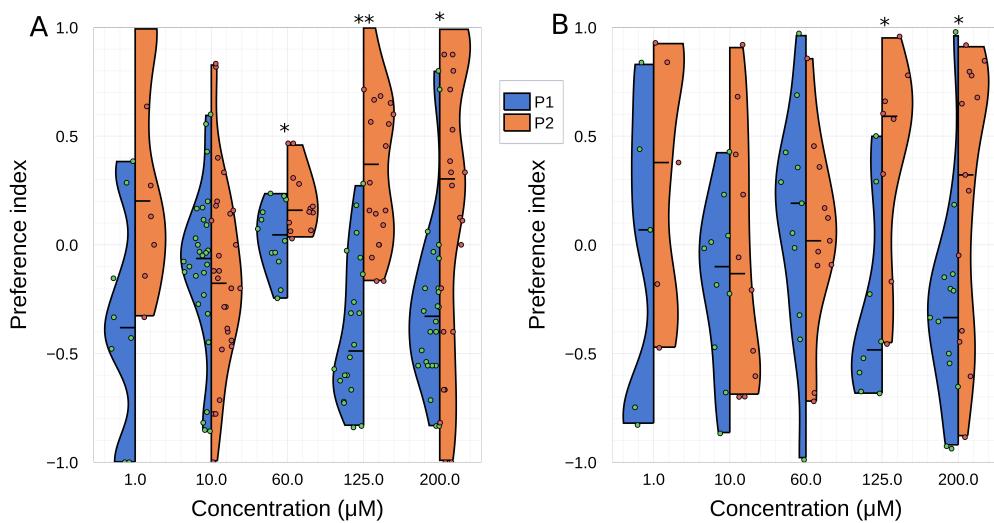


Figure G.3: ATP: preference index for juveniles **A.** Event-based preference index, one point is one fish for one independent analysis (total of $2N$ points). **B.** Time-based preference index, one point is one fish.

Bibliography

- [1] Airbus, ATTOL: Autonomous Taxiing, Take-Off and Landing, 2020. <https://www.airbus.com/newsroom/stories/autonomy-aerial-mobility.html>.
- [2] R. C. Luo, R. Mullen, and D. E. Wessell, “An adaptive robotic tracking system using optical flow,” in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pp. 568–573, IEEE, 1988.
- [3] A. I. Dell, J. A. Bender, K. Branson, I. D. Couzin, G. G. de Polavieja, L. P. Noldus, A. Pérez-Escudero, P. Perona, A. D. Straw, M. Wikelski, et al., “Automated image-based tracking and its application in ecology,” *Trends in ecology & evolution*, vol. 29, no. 7, pp. 417–428, 2014.
- [4] B. Risse, D. Berh, N. Otto, C. Klämbt, and X. Jiang, “Fimtrack: An open source tracking and locomotion analysis software for small animals,” *PLoS computational biology*, vol. 13, no. 5, p. e1005530, 2017.
- [5] R. R. Juang, A. Levchenko, and P. Burlina, “Tracking cell motion using gm-phd,” in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1154–1157, IEEE, 2009.
- [6] M. A. A. Dewan, M. O. Ahmad, and M. Swamy, “Tracking biological cells in time-lapse microscopy: An adaptive technique combining motion and topological features,” *IEEE transactions on biomedical engineering*, vol. 58, no. 6, pp. 1637–1647, 2011.

- 1892 [7] J. Jeong, N. J. Frohberg, E. Zhou, T. Sulchek, and P. Qiu, “Accurately
1893 tracking single-cell movement trajectories in microfluidic cell sort-
1894 ing devices,” *PloS one*, vol. 13, no. 2, p. e0192463, 2018.
- 1895 [8] A. Bricard, J.-B. Caussin, N. Desreumaux, O. Dauchot, and D. Barto-
1896 tolo, “Emergence of macroscopic directed motion in populations
1897 of motile colloids,” *Nature*, vol. 503, no. 7474, pp. 95–98, 2013.
- 1898 [9] I. Ali and M. N. Dailey, “Multiple human tracking in high-density
1899 crowds,” *Image and vision computing*, vol. 30, no. 12, pp. 966–977,
1900 2012.
- 1901 [10] A. Treptow and A. Zell, “Real-time object tracking for soccer-robots
1902 without color information,” *Robotics and Autonomous Systems*,
1903 vol. 48, no. 1, pp. 41–48, 2004.
- 1904 [11] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *Acm
1905 computing surveys (CSUR)*, vol. 38, no. 4, pp. 13–es, 2006.
- 1906 [12] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep
1907 learning: A review,” *IEEE transactions on neural networks and learn-
1908 ing systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- 1909 [13] Z.-M. Qian, S. H. Wang, X. E. Cheng, and Y. Q. Chen, “An effective and
1910 robust method for tracking multiple fish in video image based on
1911 fish head detection,” *BMC bioinformatics*, vol. 17, no. 1, p. 251, 2016.
- 1912 [14] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G.
1913 De Polavieja, “idtracker: tracking individuals in a group by auto-
1914 matic identification of unmarked animals,” *Nature methods*, vol. 11,
1915 no. 7, pp. 743–748, 2014.
- 1916 [15] Y.-X. Bai, S.-H. Zhang, Z. Fan, X.-Y. Liu, X. Zhao, X.-Z. Feng, and M.-Z.
1917 Sun, “Automatic multiple zebrafish tracking based on improved hog
1918 features,” *Scientific reports*, vol. 8, no. 1, pp. 1–14, 2018.
- 1919 [16] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis,
1920 and M. Bethge, “Deeplabcut: markerless pose estimation of user-

- 1921 defined body parts with deep learning,” *Nature neuroscience*, vol. 21,
1922 no. 9, pp. 1281–1289, 2018.
- 1923 [17] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. Heras, and G. G.
1924 de Polavieja, “Idtracker. ai: tracking all individuals in small or large
1925 collectives of unmarked animals,” *Nature methods*, vol. 16, no. 2,
1926 pp. 179–182, 2019.
- 1927 [18] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, P. L. Andersson, and
1928 M. Andersson, “Toxtrac: a fast and robust software for tracking or-
1929 ganisms,” *Methods in Ecology and Evolution*, vol. 9, no. 3, pp. 460–
1930 464, 2018.
- 1931 [19] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, and M. Andersson,
1932 “Toxic: an efficient algorithm to solve occlusions when tracking mul-
1933 tiple animals,” *Scientific reports*, vol. 7, no. 1, pp. 1–8, 2017.
- 1934 [20] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele,
1935 “Deepcut: A deeper, stronger, and faster multi-person pose esti-
1936 mation model,” in *European Conference on Computer Vision*, pp. 34–
1937 50, Springer, 2016.
- 1938 [21] T. Nath, A. Mathis, A. C. Chen, A. Patel, M. Bethge, and M. W. Mathis,
1939 “Using deeplabcut for 3d markerless pose estimation across species
1940 and behaviors,” *Nature protocols*, vol. 14, no. 7, pp. 2152–2176, 2019.
- 1941 [22] A. Mathis and R. Warren, “On the inference speed and video-
1942 compression robustness of deeplabcut,” *BioRxiv*, p. 457242, 2018.
- 1943 [23] TheQtCompany, *The Qt Framework*, 2020. <https://www.qt.io/>.
- 1944 [24] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software
1945 Tools*, 2000.
- 1946 [25] G. D. Evangelidis and E. Z. Psarakis, “Parametric image alignment
1947 using enhanced correlation coefficient maximization,” *IEEE Transac-
1948 tions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10,
1949 pp. 1858–1865, 2008.

- 1950 [26] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient
1951 alternative to sift or surf,” in *2011 International conference on com-*
1952 *puter vision*, pp. 2564–2571, ieee, 2011.
- 1953 [27] R. C. Bolles and M. A. Fischler, “A ransac-based approach to model
1954 fitting and its application to finding cylinders in range data.,” in *IJCAI*,
1955 vol. 1981, pp. 637–643, 1981.
- 1956 [28] M. Shahin, M. A. Babar, and L. Zhu, “Continuous integration, delivery
1957 and deployment: a systematic review on approaches, tools, chal-
1958 lenges and practices,” *IEEE Access*, vol. 5, pp. 3909–3943, 2017.
- 1959 [29] A. Wikström *et al.*, “Benefits and challenges of continuous integra-
1960 tion and delivery: A case study,” 2019.
- 1961 [30] H. W. Kuhn, “The hungarian method for the assignment problem,”
1962 *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- 1963 [31] J. Munkres, “Algorithms for the assignment and transportation prob-
1964 lems,” *Journal of the society for industrial and applied mathematics*,
1965 vol. 5, no. 1, pp. 32–38, 1957.
- 1966 [32] J. Edmonds and R. M. Karp, “Theoretical improvements in algo-
1967 rithmic efficiency for network flow problems,” *Journal of the ACM (JACM)*,
1968 vol. 19, no. 2, pp. 248–264, 1972.
- 1969 [33] N. Tomizawa, “On some techniques useful for solution of trans-
1970 portation network problems,” *Networks*, vol. 1, no. 2, pp. 173–194,
1971 1971.
- 1972 [34] R. Jonker and A. Volgenant, “A shortest augmenting path algorithm
1973 for dense and sparse linear assignment problems,” *Computing*,
1974 vol. 38, no. 4, pp. 325–340, 1987.
- 1975 [35] L. R. Yohe and P. Brand, “Evolutionary ecology of chemosensation
1976 and its role in sensory drive,” *Current zoology*, vol. 64, no. 4, pp. 525–
1977 533, 2018.

- 1978 [36] L. Bardwell, "A walk-through of the yeast mating pheromone re-
1979 sponse pathway," *Peptides*, vol. 26, no. 2, pp. 339–350, 2005.
- 1980 [37] J. G. Hildebrand and G. M. Shepherd, "Mechanisms of olfactory
1981 discrimination: converging evidence for common principles across
1982 phyla," *Annual review of neuroscience*, vol. 20, no. 1, pp. 595–631,
1983 1997.
- 1984 [38] D. A. Yarmolinsky, C. S. Zuker, and N. J. Ryba, "Common sense about
1985 taste: from mammals to insects," *Cell*, vol. 139, no. 2, pp. 234–244,
1986 2009.
- 1987 [39] T. J. Hara, *Fish chemoreception*, vol. 6. Springer Science & Business
1988 Media, 2012.
- 1989 [40] A. Hansen and K. Reutter, "Chemosensory systems in fish: structural,
1990 functional and ecological aspects," in *The senses of fish*, pp. 55–89,
1991 Springer, 2004.
- 1992 [41] T. Panier, S. Romano, R. Olive, T. Pietri, G. Sumbre, R. Candelier,
1993 and G. Debrégeas, "Fast functional imaging of multiple brain re-
1994 gions in intact zebrafish larvae using selective plane illumination
1995 microscopy," *Frontiers in neural circuits*, vol. 7, p. 65, 2013.
- 1996 [42] I. H. Bianco, A. R. Kampff, and F. Engert, "Prey capture behavior
1997 evoked by simple visual stimuli in larval zebrafish," *Frontiers in sys-
1998 tems neuroscience*, vol. 5, p. 101, 2011.
- 1999 [43] E. A. Naumann, J. E. Fitzgerald, T. W. Dunn, J. Rihel, H. Sompolinsky,
2000 and F. Engert, "From whole-brain data to functional circuit models:
2001 the zebrafish optomotor response," *Cell*, vol. 167, no. 4, pp. 947–960,
2002 2016.
- 2003 [44] S. Wolf, A. M. Dubreuil, T. Bertoni, U. L. Böhm, V. Bormuth, R. Cande-
2004 lier, S. Karpenko, D. G. Hildebrand, I. H. Bianco, R. Monasson, *et al.*,
2005 "Sensorimotor computation underlying phototaxis in zebrafish," *Na-
2006 ture communications*, vol. 8, no. 1, pp. 1–12, 2017.

- 2007 [45] R. Olive, S. Wolf, A. Dubreuil, V. Bormuth, G. Debrégeas, and R. Can-
2008 delier, "Rheotaxis of larval zebrafish: behavioral study of a multi-
2009 sensory process," *Frontiers in systems neuroscience*, vol. 10, p. 14,
2010 2016.
- 2011 [46] M. Haesemeyer, A. F. Schier, and F. Engert, "Convergent temperature
2012 representations in artificial and biological neural networks," *Neuro-
2013рон*, vol. 103, no. 6, pp. 1123–1134, 2019.
- 2014 [47] M. Ichikawa and K. Ueda, "Fine structure of the olfactory epithelium
2015 in the goldfish, carassius auratus," *Cell and tissue research*, vol. 183,
2016 no. 4, pp. 445–455, 1977.
- 2017 [48] A. Hansen and B. S. Zielinski, "Diversity in the olfactory epithelium
2018 of bony fishes: development, lamellar arrangement, sensory neu-
2019 ron cell types and transduction components," *Journal of neurocy-
2020 tology*, vol. 34, no. 3-5, pp. 183–208, 2005.
- 2021 [49] G. Ahuja, S. B. Nia, V. Zapilko, V. Shiriagin, D. Kowatschew, Y. Oka,
2022 and S. I. Korschning, "Kappe neurons, a novel population of olfactory
2023 sensory neurons," *Scientific reports*, vol. 4, no. 1, pp. 1–8, 2014.
- 2024 [50] N. Wakisaka, N. Miyasaka, T. Koide, M. Masuda, T. Hiraki-Kajiyama,
2025 and Y. Yoshihara, "An adenosine receptor for olfaction in fish," *Cur-
2026 rent Biology*, vol. 27, no. 10, pp. 1437–1447, 2017.
- 2027 [51] N. Miyasaka, K. Morimoto, T. Tsubokawa, S.-i. Higashijima,
2028 H. Okamoto, and Y. Yoshihara, "From the olfactory bulb to
2029 higher brain centers: genetic visualization of secondary olfactory
2030 pathways in zebrafish," *Journal of Neuroscience*, vol. 29, no. 15,
2031 pp. 4756–4767, 2009.
- 2032 [52] A. A. Nikonov and J. Caprio, "Electrophysiological evidence for a
2033 chemotopy of biologically relevant odors in the olfactory bulb of
2034 the channel catfish," *Journal of neurophysiology*, vol. 86, no. 4,
2035 pp. 1869–1876, 2001.

- 2036 [53] A. A. Nikonorov, T. E. Finger, and J. Caprio, "Beyond the olfactory bulb:
2037 an odotopic map in the forebrain," *Proceedings of the National*
2038 *Academy of Sciences*, vol. 102, no. 51, pp. 18688–18693, 2005.
- 2039 [54] A. Hansen and Z. Eckart, "The peripheral olfactory organ of the
2040 zebrafish, *danio rerio*: an ultrastructural study," *Chemical senses*,
2041 vol. 23, no. 1, pp. 39–48, 1998.
- 2042 [55] F. Kermen, L. M. Franco, C. Wyatt, and E. Yaksi, "Neural circuits medi-
2043 ating olfactory-driven behavior in fish," *Frontiers in neural circuits*,
2044 vol. 7, p. 62, 2013.
- 2045 [56] R. Friedrich, E. Yaksi, B. Judkewitz, and M. Wiechert, "Processing of
2046 odor representations by neuronal circuits in the olfactory bulb," *Annals of the New York Academy of Sciences*, vol. 1170, no. 1, pp. 293–
2047 297, 2009.
- 2048 [57] O. R. Braubach, A. Fine, and R. P. Croll, "Distribution and functional
2049 organization of glomeruli in the olfactory bulbs of zebrafish (*danio*
2050 *rerio*)," *Journal of comparative neurology*, vol. 520, no. 11, pp. 2317–
2051 2339, 2012.
- 2052 [58] S. Serizawa, K. Miyamichi, and H. Sakano, "One neuron–one receptor
2053 rule in the mouse olfactory system," *TRENDS in Genetics*, vol. 20,
2054 no. 12, pp. 648–653, 2004.
- 2055 [59] A. L. Barth, J. C. Dugas, and J. Ngai, "Noncoordinate expression of
2056 odorant receptor genes tightly linked in the zebrafish genome,"
2057 *Neuron*, vol. 19, no. 2, pp. 359–369, 1997.
- 2058 [60] F. Weth, W. Nadler, and S. Korschning, "Nested expression domains
2059 for odorant receptors in zebrafish olfactory epithelium," *Proceed-
2060 *ings of the National Academy of Sciences**, vol. 93, no. 23, pp. 13321–
2061 13326, 1996.
- 2062 [61] Y. Sato, N. Miyasaka, and Y. Yoshihara, "Hierarchical regulation of
2063 odorant receptor gene choice and subsequent axonal projection

- 2065 of olfactory sensory neurons in zebrafish," *Journal of Neuroscience*,
2066 vol. 27, no. 7, pp. 1606–1615, 2007.
- 2067 [62] Y. Sato, N. Miyasaka, and Y. Yoshihara, "Mutually exclusive glomerular
2068 innervation by two distinct types of olfactory sensory neurons
2069 revealed in transgenic zebrafish," *Journal of Neuroscience*, vol. 25,
2070 no. 20, pp. 4889–4897, 2005.
- 2071 [63] R. W. Friedrich and S. I. Korsching, "Chemotopic, combinatorial, and
2072 noncombinatorial odorant representations in the olfactory bulb
2073 revealed using a voltage-sensitive axon tracer," *Journal of Neuro-
2074 science*, vol. 18, no. 23, pp. 9977–9988, 1998.
- 2075 [64] S. H. Fuss and S. I. Korsching, "Odorant feature detection: activity
2076 mapping of structure response relationships in the zebrafish olfac-
2077 tory bulb," *Journal of Neuroscience*, vol. 21, no. 21, pp. 8396–8407,
2078 2001.
- 2079 [65] S. Korsching, "Odor maps in the brain: spatial aspects of odor repre-
2080 sentation in sensory surface and olfactory bulb," *Cellular and Molec-
2081 ular Life Sciences CMLS*, vol. 58, no. 4, pp. 520–530, 2001.
- 2082 [66] A. Hansen and E. Zeiske, "Development of the olfactory organ in
2083 the zebrafish, *brachydanio rerio*," *Journal of Comparative Neurol-
2084 ogy*, vol. 333, no. 2, pp. 289–300, 1993.
- 2085 [67] N. Miyasaka, A. A. Wanner, J. Li, J. Mack-Bucher, C. Genoud, Y. Yoshi-
2086 hara, and R. W. Friedrich, "Functional development of the olfactory
2087 system in zebrafish," *Mechanisms of development*, vol. 130, no. 6-8,
2088 pp. 336–346, 2013.
- 2089 [68] B. Kapoor, H. E. Evans, and E. Pevzner, "The gustatory system in fish,"
2090 in *Advances in marine biology*, vol. 13, pp. 53–108, Elsevier, 1976.
- 2091 [69] L. Fishelson, Y. Delarea, and A. Zverdling, "Taste bud form and dis-
2092 tribution on lips and in the oropharyngeal cavity of cardinal fish
2093 species (apogonidae, teleostei), with remarks on their dentition,"
2094 *Journal of Morphology*, vol. 259, no. 3, pp. 316–327, 2004.

- 2095 [70] K. Reutter, F. Boudriot, and M. Witt, "Heterogeneity of fish taste
2096 bud ultrastructure as demonstrated in the holosteans *amia calva*
2097 and *lepisosteus oculatus*," *Philosophical Transactions of the Royal*
2098 *Society of London. Series B: Biological Sciences*, vol. 355, no. 1401,
2099 pp. 1225–1228, 2000.
- 2100 [71] K. Reutter, "Ultrastructure of taste buds in the australian lungfish,
2101 *neoceratodus forsteri* (dipnoi)," *Chem. Senses*, vol. 16, p. 404, 1991.
- 2102 [72] K. Reutter, *Taste organ in the bullhead (Teleostei)*. Springer Science
2103 & Business Media, 2012.
- 2104 [73] A. Hansen, K. Reutter, and E. Zeiske, "Taste bud development in the
2105 zebrafish, *danio rerio*," *Developmental dynamics: an official publi-*
2106 *cation of the American Association of Anatomists*, vol. 223, no. 4,
2107 pp. 483–496, 2002.
- 2108 [74] Y. Ohkubo, M. Masubuchi, K. Fujioka, Y. Tomita, T. Matsushita,
2109 K. Ohsuga, and T. Marui, "Distribution and morphological features of
2110 taste buds in the zebrafish, *danio rerio*," *Journal of Oral Biosciences*,
2111 vol. 47, no. 1, pp. 77–82, 2005.
- 2112 [75] J. Yáñez, Y. Souto, L. Piñeiro, M. Folgueira, and R. Anadón, "Gustatory
2113 and general visceral centers and their connections in the brain of
2114 adult zebrafish: a carbocyanine dye tract-tracing study," *Journal of*
2115 *Comparative Neurology*, vol. 525, no. 2, pp. 333–362, 2017.
- 2116 [76] K. Kotrschal, W.-D. Krautgartner, and A. Hansen, "Ontogeny of the
2117 solitary chemosensory cells in the zebrafish, *danio rerio*," *Chemical*
2118 *senses*, vol. 22, no. 2, pp. 111–118, 1997.
- 2119 [77] O. B. Stabell, "Olfactory control of homing behaviour in salmonids,"
2120 in *Fish chemoreception*, pp. 249–270, Springer, 1992.
- 2121 [78] D. Hasler Arthur and A. T. Scholz, "Olfactory imprinting and homing
2122 in salmon. investigations in the mechanism of the imprinting pro-
cess," 1983.

- 2124 [79] D. Pavlov, A. Kasumyan, *et al.*, “Sensory principles of the feeding
2125 behavior of fishes,” *J. Ichthyol.*, vol. 30, no. 6, pp. 77–93, 1990.
- 2126 [80] J. Atema, “Chemical senses, chemical signals and feeding behavior
2127 in fishes,” *Fish Behaviour and its Use in the Capture and Culture of*
2128 *Fishes*, pp. 57–101, 1980.
- 2129 [81] W. Bateson, “The sense organs and perceptions of fishes; with re-
2130 marks on the supply of bait,” *Journal of the Marine Biological Asso-*
2131 *ciation of the United Kingdom*, vol. 1, no. 3, pp. 225–256, 1890.
- 2132 [82] W. N. TAVolga, “Visual, chemical and sound stimuli as cues in the sex
2133 discriminatory behavior of the gobiid fish *bathygobius soporator*,”
2134 *Zoologica*, vol. 41, pp. 49–64, 1956.
- 2135 [83] J. De Bruin, M. VAN DER MECHÉ-JACOBI, J. Segaar, and A. Van
2136 Der Meche, “Influence of chemical receptivity on reproductive be-
2137 haviour of the male three-spined stickleback (*gasterosteus aculea-*
2138 *tus l.*)”, *Behaviour*, vol. 86, no. 1-2, pp. 100–165, 1983.
- 2139 [84] K. v. Frisch, “Über einen schreckstoff der fischhaut und seine bio-
2140 logische bedeutung,” *Zeitschrift für vergleichende Physiologie*, vol. 29,
2141 no. 1-2, pp. 46–145, 1942.
- 2142 [85] N. Speedie and R. Gerlai, “Alarm substance induced behavioral
2143 responses in zebrafish (*danio rerio*),” *Behavioural brain research*,
2144 vol. 188, no. 1, pp. 168–177, 2008.
- 2145 [86] K. Doving and S. Lastein, “The alarm reaction in fishes-odorants,
2146 modulations of responses, neural pathways,” *Annals of the New*
2147 *York Academy of Sciences*, vol. 1170, no. 1, pp. 413–423, 2009.
- 2148 [87] S. Cassar, I. Adatto, J. L. Freeman, J. T. Gamse, I. Iturria, C. Lawrence,
2149 A. Muriana, R. T. Peterson, S. Van Cruchten, and L. I. Zon, “Use of
2150 zebrafish in drug discovery toxicology,” *Chemical research in toxici-*
2151 *cology*, vol. 33, no. 1, pp. 95–118, 2019.

- 2152 [88] E. W. Klee, H. Schneider, K. J. Clark, M. A. Cousin, J. O. Ebbert, W. M.
2153 Hooten, V. M. Karpyak, D. O. Warner, and S. C. Ekker, "Zebrafish: a
2154 model for the study of addiction genetics," *Human genetics*, vol. 131,
2155 no. 6, pp. 977–1008, 2012.
- 2156 [89] Y.-J. Dai, Y.-F. Jia, N. Chen, W.-P. Bian, Q.-K. Li, Y.-B. Ma, Y.-L. Chen, and
2157 D.-S. Pei, "Zebrafish as a model system to study toxicology," *Environmental
2158 toxicology and chemistry*, vol. 33, no. 1, pp. 11–17, 2014.
- 2159 [90] P. Mathur, B. Lau, and S. Guo, "Conditioned place preference behav-
2160 ior in zebrafish," *Nature protocols*, vol. 6, no. 3, pp. 338–345, 2011.
- 2161 [91] R. Blaser and D. Vira, "Experiments on learning in zebrafish (*danio
2162 rerio*): a promising model of neurocognitive function," *Neuroscience & Biobehavioral
2163 Reviews*, vol. 42, pp. 224–231, 2014.
- 2164 [92] A. D. Collier and D. J. Echevarria, "The utility of the zebrafish model
2165 in conditioned place preference to assess the rewarding effects of
2166 drugs," *Behavioural pharmacology*, vol. 24, no. 5 and 6, pp. 375–383,
2167 2013.
- 2168 [93] T. M. Tzschenkentke, "Review on cpp: Measuring reward with the condi-
2169 tioned place preference (cpp) paradigm: update of the last decade,"
2170 *Addiction biology*, vol. 12, no. 3-4, pp. 227–462, 2007.
- 2171 [94] T. Darland and J. E. Dowling, "Behavioral screening for cocaine
2172 sensitivity in mutagenized zebrafish," *Proceedings of the National
2173 Academy of Sciences*, vol. 98, no. 20, pp. 11691–11696, 2001.
- 2174 [95] P. Mathur, M. A. Berberoglu, and S. Guo, "Preference for ethanol in
2175 zebrafish following a single exposure," *Behavioural brain research*,
2176 vol. 217, no. 1, pp. 128–133, 2011.
- 2177 [96] J. Ninkovic, A. Folchert, Y. V. Makhankov, S. C. Neuhauss, I. Sill-
2178 aber, U. Straehle, and L. Bally-Cuif, "Genetic identification of ache
2179 as a positive modulator of addiction to the psychostimulant d-
2180 amphetamine in zebrafish," *Journal of neurobiology*, vol. 66, no. 5,
2181 pp. 463–475, 2006.

- 2182 [97] J. Ninkovic and L. Bally-Cuif, "The zebrafish as a model system for
2183 assessing the reinforcing properties of drugs of abuse," *Methods*,
2184 vol. 39, no. 3, pp. 262–274, 2006.
- 2185 [98] D. Braida, V. Limonta, S. Pegorini, A. Zani, C. Guerini-Rocco, E. Gori,
2186 and M. Sala, "Hallucinatory and rewarding effect of salvinorin a
2187 in zebrafish: κ -opioid and cb 1-cannabinoid receptor involvement,"
2188 *Psychopharmacology*, vol. 190, no. 4, pp. 441–448, 2007.
- 2189 [99] X. Kedikian, M. P. Faillace, and R. Bernabeu, "Behavioral and molecu-
2190 lar analysis of nicotine-conditioned place preference in zebrafish,"
2191 *PLoS One*, vol. 8, no. 7, p. e69453, 2013.
- 2192 [100] A. J. Rennekamp and R. T. Peterson, "15 years of zebrafish chemical
2193 screening," *Current opinion in chemical biology*, vol. 24, pp. 58–70,
2194 2015.
- 2195 [101] V. Sallinen, V. Torkko, M. Sundvik, I. Reenilä, D. Khrustalyov, J. Kaslin,
2196 and P. Panula, "Mptp and mpp+ target specific aminergic cell popu-
2197 lations in larval zebrafish," *Journal of neurochemistry*, vol. 108, no. 3,
2198 pp. 719–731, 2009.
- 2199 [102] J. Rihel, D. A. Prober, A. Arvanites, K. Lam, S. Zimmerman, S. Jang,
2200 S. J. Haggarty, D. Kokel, L. L. Rubin, R. T. Peterson, *et al.*, "Zebrafish
2201 behavioral profiling links drugs to biological targets and rest/wake
2202 regulation," *Science*, vol. 327, no. 5963, pp. 348–351, 2010.
- 2203 [103] D. Kokel, J. Bryan, C. Laggner, R. White, C. Y. J. Cheung, R. Ma-
2204 teus, D. Healey, S. Kim, A. A. Werdich, S. J. Haggarty, *et al.*, "Rapid
2205 behavior-based identification of neuroactive small molecules in
2206 the zebrafish," *Nature chemical biology*, vol. 6, no. 3, pp. 231–237,
2207 2010.
- 2208 [104] M. J. Winter, W. S. Redfern, A. J. Hayfield, S. F. Owen, J.-P. Valentin,
2209 and T. H. Hutchinson, "Validation of a larval zebrafish locomotor
2210 assay for assessing the seizure liability of early-stage develop-
2211 ment drugs," *Journal of pharmacological and toxicological meth-*
2212 *ods*, vol. 57, no. 3, pp. 176–187, 2008.

- 2213 [105] T. C. Farrell, C. L. Cario, C. Milanese, A. Vogt, J.-H. Jeong, and E. A. Burton, “Evaluation of spontaneous propulsive movement as a screening
2214 tool to detect rescue of parkinsonism phenotypes in zebrafish
2215 models,” *Neurobiology of disease*, vol. 44, no. 1, pp. 9–18, 2011.
- 2216
- 2217 [106] Q. Shen, L. Truong, M. T. Simonich, C. Huang, R. L. Tanguay, and Q. Dong, “Rapid well-plate assays for motor and social behaviors
2218 in larval zebrafish,” *Behavioural Brain Research*, p. 112625, 2020.
- 2219
- 2220 [107] S. Schnörr, P. Steenbergen, M. Richardson, and D. Champagne, “Measuring thigmotaxis in larval zebrafish,” *Behavioural brain research*,
2221 vol. 228, no. 2, pp. 367–374, 2012.
- 2222
- 2223 [108] S. D. Pelkowski, M. Kapoor, H. A. Richendrfer, X. Wang, R. M. Col-
2224 will, and R. Creton, “A novel high-throughput imaging system for
2225 automated analyses of avoidance behavior in zebrafish larvae,” *Be-
2226 havioural brain research*, vol. 223, no. 1, pp. 135–144, 2011.
- 2227
- 2228 [109] S. Krishnan, A. S. Mathuru, C. Kibat, M. Rahman, C. E. Lupton, J. Stew-
2229 art, A. Claridge-Chang, S.-C. Yen, and S. Jesuthasan, “The right dorsal
2230 habenula limits attraction to an odor in zebrafish,” *Current Biology*,
vol. 24, no. 11, pp. 1167–1175, 2014.
- 2231
- 2232 [110] A. Hussain, L. R. Saraiva, D. M. Ferrero, G. Ahuja, V. S. Krishna, S. D.
2233 Liberles, and S. I. Korsching, “High-affinity olfactory receptor for
2234 the death-associated odor cadaverine,” *Proceedings of the National
Academy of Sciences*, vol. 110, no. 48, pp. 19579–19584, 2013.
- 2235
- 2236 [111] G. D. Readman, S. F. Owen, J. C. Murrell, and T. G. Knowles, “Do fish
2237 perceive anaesthetics as aversive?” *PLoS One*, vol. 8, no. 9, p. e73773,
2013.
- 2238
- 2239 [112] M. S. Abreu, A. C. V. Giacomini, D. Gusso, J. G. Rosa, G. Koakoski,
2240 F. Kalichak, R. Idalêncio, T. A. Oliveira, H. H. Barcellos, C. D. Bonan,
2241 *et al.*, “Acute exposure to waterborne psychoactive drugs attract
zebrafish,” *Comparative Biochemistry and Physiology Part C: Toxi-
cology & Pharmacology*, vol. 179, pp. 37–43, 2016.
- 2242

- 2243 [113] M. S. Abreu, A. C. V. Giacomini, D. Gusso, G. Koakoski, T. A. Oliveira,
2244 A. Marqueze, R. E. Barreto, and L. J. Barcellos, “Behavioral responses
2245 of zebrafish depend on the type of threatening chemical cues,” *Journal*
2246 of Comparative Physiology A, vol. 202, no. 12, pp. 895–901, 2016.
- 2247 [114] F. Kermen, L. Darnet, C. Wiest, F. Palumbo, J. Bechert, O. Uslu, and
2248 E. Yaksi, “Stimulus-specific behavioral responses of zebrafish to
2249 a large range of odors exhibit individual variability,” *BMC biology*,
2250 vol. 18, no. 1, pp. 1–16, 2020.
- 2251 [115] A. O. Kasumyan and K. B. DÖving, “Taste preferences in fishes,” *Fish*
2252 and fisheries, vol. 4, no. 4, pp. 289–347, 2003.