



HAL
open science

Assessing chemical perception of the young zebrafish

Benjamin Gallois

► **To cite this version:**

Benjamin Gallois. Assessing chemical perception of the young zebrafish. Biophysics. Sorbonne université, 2021. English. NNT: . tel-03243224

HAL Id: tel-03243224

<https://hal.science/tel-03243224>

Submitted on 31 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Thèse de doctorat

Sorbonne Université

École doctorale n°564 : Physique en Île-de-France

Assessing chemical perception of the young zebrafish

réalisée au Laboratoire Jean Perrin par

Benjamin Gallois

soutenue le 31 Mars 2021

devant le jury composé de

M. JOLY Jean-Stéphane

M. BOURDIEU Laurent

Mme WYART Claire

Mme HONG Elim

M. THIRIA Benjamin

M. DEL BENE Filippo

M. CANDELIER Raphaël

Referee

Referee

Examinator

Examinator

Invited

Invited

Thesis director

Abstract

Tracking several objects from video recordings is a complicated image processing task but essential in many academic fields. We compiled a database of two-dimensional movies from very different biological and physical systems, spanning a wide range of scales and dynamics, and developed a general-purpose tracking software called FastTrack. It can handle a changing number of deformable objects. Furthermore, we introduce the probability of incursion, a new measure of a movie's trackability that does not require the ground truth trajectories. We demonstrated that FastTrack is orders of magnitudes faster than state-of-the-art tracking algorithms with a comparable tracking accuracy. A user and developer documentation is available, and the software is distributed under a GNU GPLv3 license. Chemical perception mediates several essential behaviors in fish, like mating and feeding. However, there is much to understand on how the fish process chemical stimuli. We build two experimental setups: Dual, a high-throughput setup capable of assessing young zebrafish chemical preference, and The Tropical River to simulate realistic flows that fish are susceptible to encounter in nature. Dual is scalable, open-source, and can be built for less than 2 000 euros. Using it, we showed a clear repulsion to citric acid from fish up to 2 weeks old and showed that ATP's presentation to 2 weeks old fish was first repulsive and then attractive.

Résumé

Le suivi d'objets à partir d'enregistrements vidéo est une tâche de traitement d'image compliquée mais essentielle dans de nombreux domaines universitaires. Nous avons compilé une base de données de films bidimensionnels provenant de systèmes biologiques et physiques très différents, couvrant une large gamme d'échelles et de dynamiques, et avons développé un logiciel de suivi polyvalent appelé FastTrack. Il peut traiter un nombre variable d'objets déformables. En outre, nous introduisons la probabilité d'incursion, une nouvelle mesure de la difficulté d'analyse d'un film qui ne nécessite pas les trajectoires. Nous avons démontré que FastTrack est plus rapide de plusieurs ordres de grandeur que ce qu'il se fait de mieux dans le domaine. Une documentation utilisateur et développeur est disponible, et le logiciel est distribué sous une licence GNU GPLv3. La perception chimique intervient dans plusieurs comportements essentiels des poissons, comme l'accouplement et l'alimentation. Cependant, il en reste beaucoup à comprendre sur la façon dont les poissons traitent les stimuli chimiques. Nous avons mis en place deux dispositifs expérimentaux : Dual, un dispositif à haut débit capable d'évaluer la préférence chimique des jeunes poissons-zèbres, et The Tropical River pour simuler les flux réalistes que les poissons sont susceptibles de rencontrer dans la nature. Dual est évolutif, open-source, et peut être construit pour moins de 2 000 euros. En l'utilisant, nous avons montré une nette répulsion à l'acide citrique des poissons jusqu'à l'âge de 2 semaines et avons montré que la présentation de l'ATP aux poissons de 2 semaines était d'abord répulsive puis attractive.

Résumé long

FastTrack un logiciel de suivi généraliste

Introduction

Le suivi d'objets depuis des enregistrements vidéos est un problème qui a gagné en popularité, tant dans l'industrie que dans le milieu académique. Citons par exemple le projet ATTOL d'Airbus qui permet le roulage, le décollage et l'atterrissage d'un avion en se basant uniquement sur de l'analyse d'images. Dans le milieu académique, le suivi sur vidéos est très utilisé en biologie et en écologie. Il permet de suivre les animaux dans leur environnement sans avoir besoin de les marquer invasivement.

On se concentrera dans cette thèse sur le suivi d'objets multiples (MOT) qui regroupe la majorité des applications scientifiques. Le suivi d'objets multiples est un problème qui consiste à détecter et à garder l'identité des objets tout au long d'un enregistrement vidéo. C'est un problème complexe sous plusieurs aspects : la détection des objets peut être compliquée si par exemple ceux-ci disparaissent derrière des objets du décor, de plus la qualité des images influence grandement la détection. Les objets peuvent sortir et rentrer dans le champ de vue où se superposer ce qui complique le maintien de l'identité de chaque objet.

On distingue deux grandes classes d'algorithmes permettant de résoudre ces problèmes. Le premier utilise les paramètres cinématiques de l'objet ce qui permet ainsi de prédire et retrouver l'identité des objets d'une image sur l'autre. Très rapide, cette classe d'algorithmes souffre d'un problème majeur, la propagation des erreurs. Si une erreur est commise sur une image, elle se propagera jusqu'à la fin du film. La deuxième classe d'algorithmes utilise une "carte d'identité" extraite pour chaque objet. Cela permet de contourner le problème de propagation des erreurs au prix d'un temps de calcul très élevé.

Plusieurs logiciels de suivi existent. On peut citer Ethovision XT, Any-maze et ToxTrack pour les logiciels propriétaires. Les deux premiers sont livrés clef-en-mains mais coûtent cher ce qui peut être un frein pour cer-

tains laboratoires. Ces logiciels sont closed-source, c'est-à-dire qu'on ne peut ni les modifier, ni savoir exactement comment ils fonctionnent, ils ne pourront donc pas être adaptés pour un projet particulier. Dans les logiciels open-sources, on peut citer DeepLabCut, idTrackerai et idtracker. Les deux premiers utilisent le machine learning pour effectuer le suivi. Dans les trois cas, ces logiciels nécessitent des ordinateurs puissants et l'analyse est en général longue, l'installation est complexe et nécessite de bonnes connaissances en informatique.

Dataset

Nous avons en premier lieu regroupé divers films pouvant servir de test pour les algorithmes de suivi. Ce dataset nommé The Two Dimensional Tracking Dataset TD^2 regroupe 41 films de plus de 7 espèces animales allant du poisson à la drosophile, des particules actives, des gouttes microfluidiques et des objets macroscopiques comme des voitures et des joueurs d'ultimate.

FastTrack

Pour répondre au problème du suivi d'objets multiples, nous avons développés un logiciel nommé FastTrack. Ce logiciel est basé sur une approche inédite du suivi : au lieu de développer un système très spécifique qui ne sera utilisable que sur un très petit nombre de systèmes, FastTrack implémente un algorithme de suivi généraliste utilisable sur une grande variété de systèmes, un outil ergonomique de gestions des erreurs est ensuite proposé pour que l'utilisateur puisse corriger les trajectoires après le suivi.

Le flux de traitement de FastTrack peut être divisé en 3 étapes. La première consiste à détecter les objets. Ceci est fait en calculant et soustrayant le fond aux images puis en appliquant un seuil. FastTrack intègre les opérations d'analyses d'images usuelles pour faciliter la détection. Les objets sont ensuite triés par taille ce qui permet d'écarter les artefacts. Dans une deuxième étape, les objets sont assignés d'une image sur l'autre

ce qui permet de garder leurs identités. Ceci est fait en calculant une fonction de coût et en la minimisant pour trouver l'assignation optimale. La fonction de coût comprend le déplacement, le changement d'orientation, de taille et de périmètre des objets entre deux images successives et peut être réglée par l'utilisateur au moyen d'un ensemble de paramètres de normalisation. Deux autres paramètres de seuil permettent de définir une mémoire et une taille maximale d'assignation. La troisième et dernière étape est la correction manuelle des erreurs qui se fait dans un environnement interactif et ergonomique.

FastTrack permet grâce à cette technique d'être applicable sur un grand nombre de systèmes. Les paramètres de normalisation permettent de l'adapter à n'importe quelle dynamique d'objets. Un jeu de paramètres neutres peut être automatiquement trouvé par le logiciel pour aider l'utilisateur à obtenir un suivi le plus optimal possible. Contrairement aux logiciels existants, FastTrack peut tracker des films à nombre d'objets variables, c'est-à-dire dont les objets peuvent disparaître puis réapparaître ou de nouveaux objets entrer dans le champ de vision. Nous avons montré que les performances de FastTrack sont aussi bonnes que les logiciels existants. De plus, notre approche permet à l'utilisateur de gagner du temps sur la plupart des projets, le temps de correction manuel étant en général plus faible que celui nécessaire pour faire un suivi directement sans erreur avec un algorithme plus coûteux en temps de calculs.

Nous avons montré comment nous pouvons classer le dataset en utilisant la probabilité d'incursion. Une incursion survient lorsque l'objet à tracker sort de sa cellule de Voronoï. La probabilité d'incursion peut être définie en utilisant uniquement les propriétés géométriques de l'objet et la distribution des déplacements. Les films peuvent alors être classés suivant leur difficulté de suivi grâce à cette probabilité et ainsi estimer le nombre de corrections manuelles qu'il faudra effectuer. Nous définissons grâce à cette probabilité un critère permettant de calculer la fréquence d'acquisition optimale qui est une question fréquente lors de la conception d'expériences.

La conception de FastTrack repose sur des bibliothèques et langages ouverts ce qui permet à n'importe qui de voir le code source et de le mo-

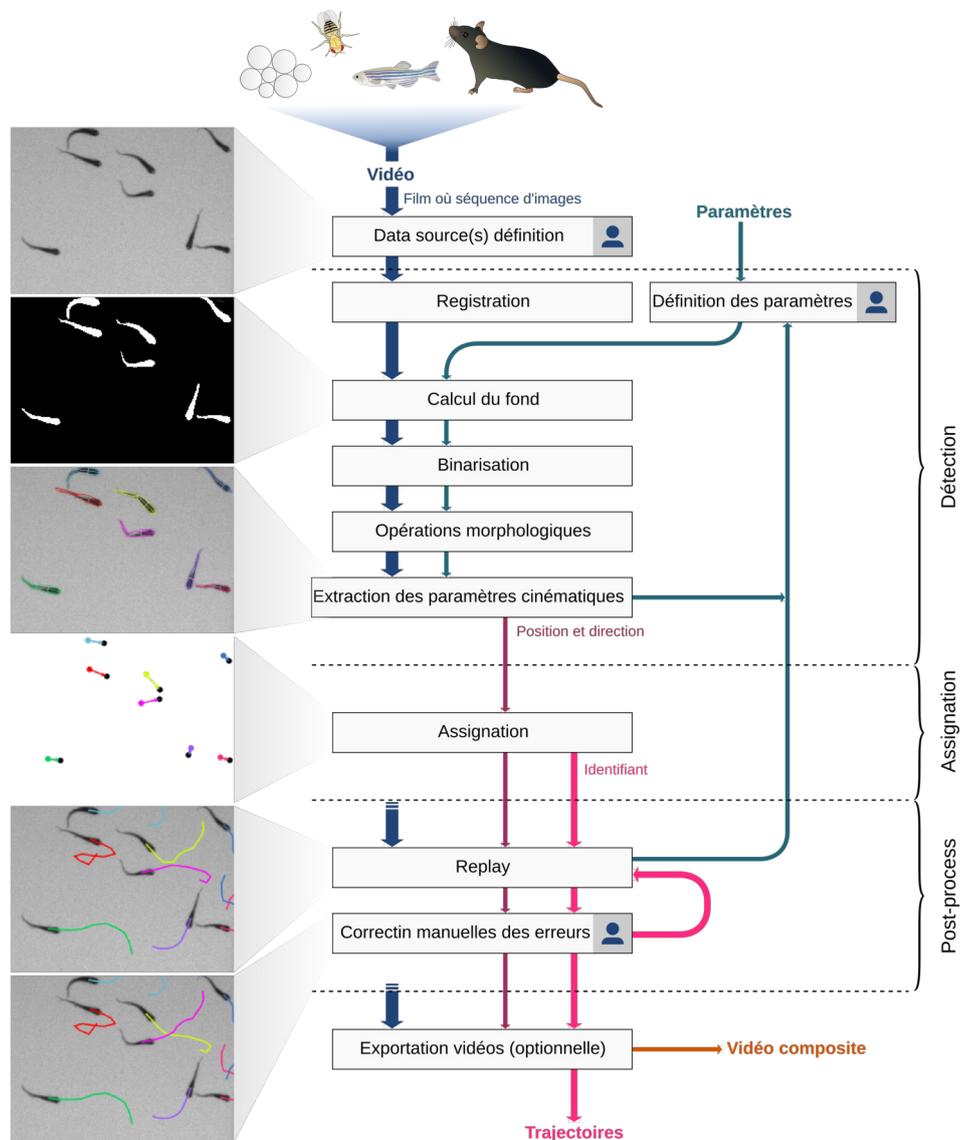


FIGURE 1 – FastTrack flux de traitement Le flux de traitement se divise en 3 étapes : la détection, l'association, et la correction. Les 👤 indiquent les étapes nécessitant l'utilisateur (film : ZFJ_001.)

difier. FastTrack est entièrement documenté et peut être intégré dans un projet déjà existant. FastTrack dispose d'un système d'intégration et de déploiement continu (CI/CD) grâce au système GitHub Actions. Il est disponible pour Linux (AppImage), MacOS et Windows et facilement installable. Un manuel d'utilisation et des tutoriels vidéos sont disponibles.

Caractérisation de la perception chimique chez le jeune poisson-zèbre

Introduction

La perception chimique est l'une des plus anciennes modalités sensorielles. Présente dans une grande variété de taxons, des unicellulaires jusqu'aux mammifères, elle est associée à des comportements nécessaires à la survie de l'espèce tels que trouver de la nourriture, se reproduire ou éviter des prédateurs. Les poissons sont baignés dans leur environnement chimique à chaque instant et sont pourvus d'organes pour percevoir et interpréter ces stimuli chimiques. Pour les poissons, la perception chimique passe par l'odorat, le goût et un sens chimique commun. Les mécanismes de perceptions ont été largement étudiés chez diverses espèces de poissons, mais peu est connu sur certains comportements complexes tels que par exemple les migrations.

Le poisson-zèbre est un modèle en pleine expansion dans le cadre des neurosciences. La larve est transparente ce qui permet d'observer l'intégralité du cerveau à l'échelle cellulaire grâce à l'image calcique à nappe de lumière. L'apparition de systèmes de réalité virtuelle pour observer le cerveau de larves effectuant des tâches a permis de mieux comprendre des comportements tels que la phototaxie, la capture de proies et la rhéotaxie. L'application de cette technique à la perception chimique nécessite quelques étapes préalables, par exemple une bonne caractérisation des produits et de la réponse comportementale qu'ils entraînent en fonction de la concentration. Pour cela, il est nécessaire d'avoir un dispositif expérimental permettant de faire varier les produits, leurs concentrations ainsi que l'âge du poisson tout en caractérisant leurs préférences. Dans un deuxième temps, un montage permettant de reproduire des écoulements réalistes sera nécessaire pour étudier la navigation par perception chimique.

Montages expérimentaux

Pour caractériser la perception chimique chez le jeune poisson-zèbre, nous avons construit deux montages expérimentaux, l'un permettant de faire un criblage des préférences des poissons à divers stimuli chimiques, l'autre permettant de recréer des écoulements réalistes pour étudier la réponse comportementale du poisson.

Dual Dual est un montage expérimental permettant de quantifier la préférence des poissons vis-à-vis d'un produit chimique de concentration parfaitement contrôlée. Il permet de séparer un compartiment où est placé le poisson en deux zones distinctes, l'un avec un produit de concentration parfaitement connue, l'autre avec de l'eau. Ceci est réalisé grâce à un flux créé par un double pousse-seringue permettant de maintenir un écoulement à volume constant dans le compartiment : quand deux seringues injectent d'un côté, deux seringues aspirent de l'autre. Un système de valves permet de remplir les seringues avec le produit choisi puis d'injecter ensuite. L'aquarium est coupé de l'environnement par une boîte et l'expérience filmée en lumière infrarouge pour éviter toutes implications d'une autre modalité sensorielle, principalement la vision. L'écoulement est visualisé par un colorant infrarouge. Ce montage expérimental est open-source, flexible, et peut être reproduit pour moins de 2 000 euros avec peu de matériel (imprimante 3D et découpe laser). Un logiciel de contrôle open-source est disponible et permet de contrôler le dispositif.

The Tropical River The Tropical River est un montage expérimental qui permet de recréer des écoulements plus réalistes auxquels sont soumis les poissons dans leur environnement naturel. Il est constitué d'un canal de 60×10×10 cm dans lequel est placé le poisson. Un écoulement laminaire contrôlé en température et en débit alimente ce canal. Un système de valves et d'injecteurs permet de créer des jets laminaires et turbulents de manière à étudier la perception chimique du poisson dans un environnement plus proche de la réalité où la perception est fragmentée.

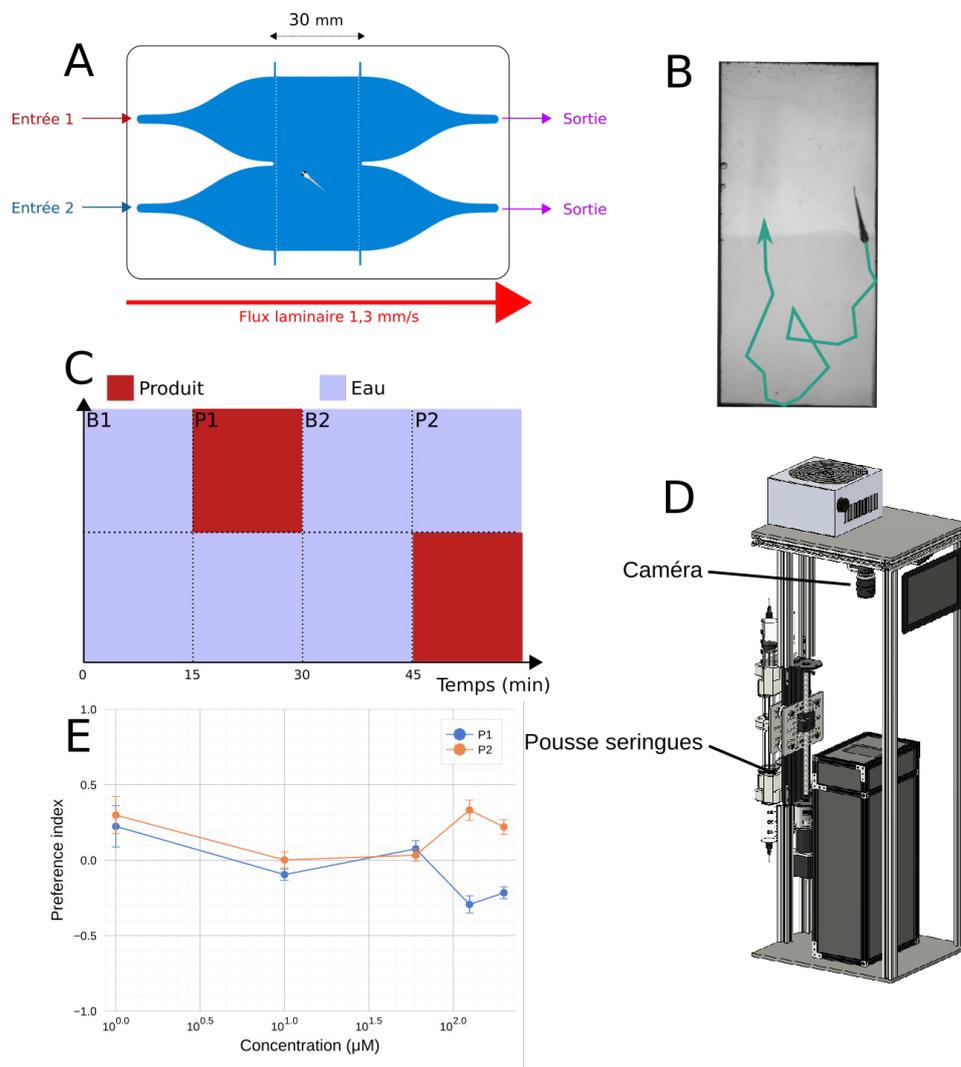


FIGURE 2 – A Schéma du compartiment où nage le poisson. **B** Image d'un écoulement typique avec en bas le produit visible à l'aide du colorant infrarouge. **C** Schéma du protocole expérimental permettant de déterminer la préférence des poissons avec un cycle contrôle B1, 2 cycles de test P1, P2 et un cycle de rinçage B2. **D** Le dispositif expérimental Dual. **E** Préférence index (temps passé dans le produit moins temps passé dans l'eau divisé par le temps total) pour l'ATP en fonction de la concentration. On remarque une répulsion au premier cycle P1 (préférence index négatif) et une attraction au cycle P2 (préférence index positif).

Résultats

Nous avons étudié la préférence de poisson-zèbre âgés de 14 jours et de 7 jours en utilisant Dual. L'expérience dure une heure durant laquelle

le poisson est soumis à un cycle avec de l'eau des deux côtés (B1) servant de contrôle, un cycle avec un produit d'un côté et de l'eau de l'autre (P1), un cycle de rinçage similaire à B1, enfin le même cycle que P1 mais en inversant les côtés (P2).

Nous nous sommes concentrés sur 5 produits : l'acide citrique et la quinine connus pour être répulsifs, l'ATP et l'adénosine connus pour être attractifs chez les poissons-zèbres adultes.

En premier lieu nous avons contrôlé que l'expérience ne contenait aucun biais et que le colorant servant à visualiser l'écoulement était bien neutre pour le poisson. Nous avons ensuite montré que les poissons-zèbres étaient repoussés par l'acide citrique, l'intensité variant en fonction de la concentration.

Nous avons trouvé un effet de répulsion à la première présentation d'ATP, puis d'attraction à la seconde présentation chez la majorité des poissons-zèbres âgés de 14 jours. Cette étude effectuée aussi chez les larves de 7 jours manque de statistique mais cet effet ne semble pas être présent. Cela indiquerait une évolution temporelle de ce phénomène absent chez les larves (7 jours), présent chez les adultes (2 mois) et une partie des juvéniles (2 semaines).

Remerciements

Je voudrais en premier lieu remercier Raphaël pour son encadrement durant ces 3.5 années, qui m'a laissé la liberté et l'autonomie dont j'avais besoin pour travailler tout en me guidant sur un sujet qui fût parfois difficile. Ainsi que pour les bons moments passé à bricoler Dual qui nous aura posé des énigmes jusqu'à la dernière semaine.

Merci à Hugo Trentesaux qui m'a présenté Raphaël et le laboratoire. Sans lui je n'aurais sûrement pas fait une thèse au LJP, ni pris autant de plaisir à construire des dispositifs expérimentaux.

Merci à tous les doctorants (et surtout ceux de la salle 417) pour la très bonne ambiance et l'aide qu'ils ont apportée durant ces 3 ans.

Je remercie l'équipe de la plateforme aquatique pour l'élevage des poissons, ainsi que toutes les personnes du laboratoire. Enfin je remercie ma famille pour leur soutien.

Contents

I	FastTrack: a general tracking software	14
1	Introduction	15
1.1	Video based tracking	15
1.2	The tracking, a not so simple problem	17
1.3	Existing software	18
1.3.1	Proprietary software	18
1.3.2	Open-source software	21
1.4	FastTrack: an original approach	24
2	Movies dataset	26
3	Design and implementation	29
3.1	Tools used	29
3.2	Implementation	31
3.2.1	Detection	31
3.2.2	Matching	36
3.2.3	Automatic tracking parameters	37

3.2.4	Manual correction	41
3.2.5	Output	41
3.3	Deployment	42
3.3.1	Continuous integration delivery	42
3.3.2	Documentation	43
4	Results	45
4.1	Performance	45
4.2	Dataset classification	47
4.3	Parameters optimization	52
5	Conclusion and perspective	55
	Appendices	57
A	FastTrack user interface preview	58
B	Voronoi diagram	60
B.1	Definition	60
B.2	Construction	61
C	Hungarian algorithm	62
C.1	Definition	62
C.2	Description	62
D	Fitting of the distribution of displacement	65

II	Assaying chemical preference of the young zebrafish	66
6	Introduction	67
6.1	The chemical perception	67
6.1.1	Olfaction	69
6.1.2	Gustation	71
6.1.3	Common chemical sense	73
6.2	Behavioral studies	74
6.2.1	Behavior	74
6.2.2	Conditioned place preference	75
6.2.3	Multi well-plate	76
6.2.4	Direct introduction	77
6.2.5	Flow	79
7	Experimental setups	82
7.1	Dual	82
7.1.1	Overview	82
7.1.2	Construction	83
7.1.3	Construction and usage	88
7.2	The Tropical River	89
7.2.1	Description	90
7.2.2	Usage and limitation	92
8	Results	94

8.1	Methods	94
8.1.1	Experiment	94
8.1.2	Analysis	96
8.2	Results	102
8.2.1	Setup characterisation	102
8.2.2	Products screening	104
9	Discussion	115
	Appendices	118
E	Dual bill of materials	119
F	Touch-and-turn behavior	122
G	Calculation	124
G.1	Mean quantities	124
G.2	Statistical tests	125
G.3	Image analysis	125
H	Kinematic parameter distributions	126
I	Markov model results	128
J	Preference index distributions	130

Part I

FastTrack: a general tracking software

Chapter 1

Introduction

Talk is cheap. Show me the code.

Linux Torvald

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 academia 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 planes 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 reduces costs, production time, and human operators' use.

In academia, the use of automated tracking, especially in biology and

ecology [3,4], is a rapidly growing field because it avoids disturbing the animals with invasive markings. Cellular motions tracking is also widely studied with very specialized algorithms developed solely for this purpose [5,6]. Other fields of science are interested in automated tracking, to name a few: microfluidic [7], active matter [8], social science [9] and robotic [10]. Automated tracking generally produces large amounts of reliable data, reduces biases and avoids long and tedious manual analyses. The latter are sometimes impossible to perform due to excessively large image collections.

Object tracking can be separated into two categories: the Single Object Tracking (SOT), where the goal is to detect a single object in a more or less complicated scene, and the Multiple Object Tracking (MOT), where the goal is to detect and track several objects. In this dissertation, we will place ourselves within the MOT framework since it is more representative of the applications usually encountered in academia. For many scientific experimental setups, the inherent difficulty of tracking can be greatly mitigated with a well-designed system. In general, the setups are designed to optimize the imaging conditions, with a fixed camera and a lighting that facilitates object detection. On the other hand, the tolerance to errors is low if one wants to produce reliable data and robust scientific conclusions. A decisive point is the algorithm's performance, which must analyze the data in a reasonable time compared to their production rate and meet the user's material and technical constraints. The ease of installation and use of the software that integrates the algorithm should not be neglected. The users brought to use these software are generally not experts in computer science and image analysis, and the software must be readily installable and usable by all.

We will first see why the tracking is still a complex problem and how we can reduce or bypass this complexity. We will then present a non-exhaustive list of existing tracking software applied to diverse scientific fields. Finally, we will present how the software we have developed for general-purpose tracking follows a different approach, and in which cases it can be useful.

1.2 The tracking, a not so simple problem

The image-based tracking of objects usually involves 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 allows for simple detection by subtracting a background image (without object) 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 sufficient 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 object 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 this makes it a good candidate for real-time tracking applications. The major disadvantage of this approach comes from the error propagation phenomenon. If

the algorithm makes an error in the assignment, it has no way to correct the error at the next step, and it propagates to the end of the analysis. The second class of algorithms is based on recognizing the object's points of interest, allowing the creation of a "fingerprint" unique to each object. That can be done using either a classical method [14,15], or using machine learning [16,17]. This technique solves the propagation of errors problem and allows objects to be tracked over time, i.e., across several unpaired videos. For example, an animal can be recognized from one day of experiments to the next, which can be very useful, especially for behavioral studies. This method requires images of sufficient quality to extract markers representative of the object. It also requires more computational resources, thus an analysis that cannot be done in real-time. However, the main limitation is the number of objects it can track. It is currently limited to about ten objects per image with classical methods before the algorithms' performance degrades significantly. The machine learning approach makes it possible to increase the number of objects at the cost of long computation time and the need to use high-performance computers.

1.3 Existing software

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

1.3.1 Proprietary software

The proprietary software presented here are closed-source. The user cannot modify the code to adapt the software to his project or check precisely how the tracking is performed. On the other hand, they do not require any computer knowledge and benefit from a support service convenient for users that do not have a lot of computer knowledge. They are an excellent alternative to other options that are sometimes difficult to implement, but their high price can be a hindrance for some users.

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

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

Any-maze Any-maze is a software developed by Stoelting Co. It is specialized in the analysis of behavioral neuroscience experiments. It directly integrates tools for standardized tests (e.g., forced swim test, fear conditioning test), allowing fully automated analysis of these experiments. It can track in real-time or from recorded videos.

Any-maze is a complete solution for creating and analyzing typical behavioral experiments. It can be purchased with the experimental setup already optimized and calibrated for the software. The Any-maze suite consists of three software packages. The tracking part is available for USD 6,495 or USD 1,500 per year. The software is available for Windows only.

ToxTrack ToxTrack [18] is a software that implements in a graphical interface the ToxId algorithm [19]. In short, the algorithm extracts objects from the background by applying a threshold. The pieces of trajectories between occlusions are divided into short and long trajectories based on a user-defined threshold time. A group of long trajectories where all individuals are observed simultaneously is then extracted. In this case, the assign-

ment is made using the Hungarian algorithm. The remaining trajectories are then assigned to the corresponding object selecting the best correlation value in a trajectory identification matrix, see Figure 1.1. This matrix contains the similarity between every two trajectory fragments based on objects' features. The authors report that ToxId is as powerful as other existing software, fast, and can track objects in real-time. A disadvantage that can be seen in this algorithm is that it only works for a constant number of animals. The algorithm's initialization requires to have at one moment t all the objects to be tracked simultaneously detectable for a user-defined time $t + dt$. The user-interface (UI) is sometimes difficult to use: the integrated tracking reviewer does not permit to correct the tracking or to replay the tracking frame by frame.

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

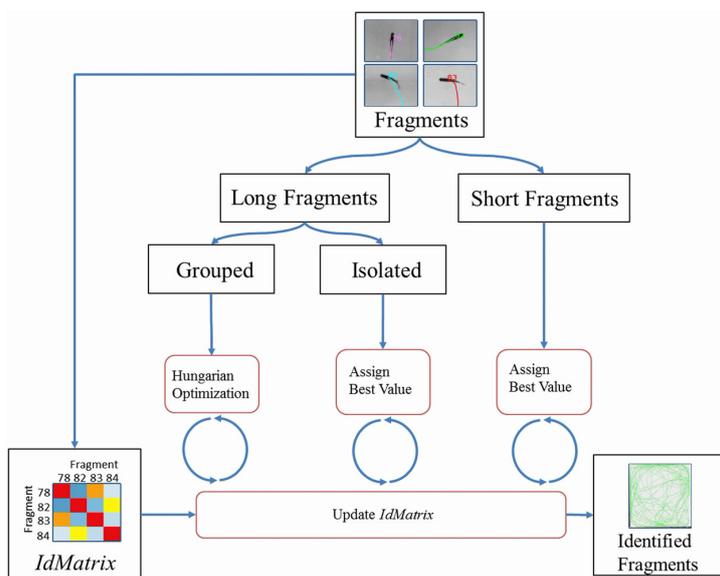


Figure 1.1 – ToxId workflow chart.

1.3.2 Open-source software

Open-source software allows the user to read, modify, and distribute the software. It is the preferred alternative to commercial software. From a scientific perspective, using open-source software increase transparency and lead to easier reproducibility of scientific results. From a development 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 recordings. It is based on the extraction of a "fingerprint" for each object, allowing a tracking 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 error-propagation 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 quality is an essential factor and must be at least 150 pixels per animal. The computation time is relatively long, in the order of 0.5 to 2 seconds per image, and requires a large amount of RAM. The installation of idTracker can be done without the need to install MATLAB thanks to the Matlab Run Time Compiler but only under Windows. Therefore, it is necessary to purchase a MATLAB license for other platforms and have minimal knowledge of the language to set up idTracker.

DeepLabCut DeepLabCut [16] is a framework that solves the so-called "pose estimation" problem, which consists of finding an object and its position, or part of an object, in an image. It can be directly related to the SOT problem if the objects to be tracked are different, for example, a right

mouse ear and a mouse nose, which can then be found on each image and then associated in the case where there is only one mouse. In the case of several similar objects to be found and associated from one image to another (MOT), this detection will have to be combined with an association step to obtain the tracking. Even if DeepLabCut answers a slightly different problem, it can, by its design, be coupled with an external association algorithm to make a tracking software.

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

DeepLabCut takes advantage of deep learning, a machine-learning algorithm that consists of training a neural network containing several layers [21]. In DeepLabCut, the network consists of several residual neural networks (ResNets) pre-trained on the ImageNet database. The network is then fine-tuned by training on images where the parts to be detected are annotated. In the end, the algorithm gives the probability of presence of the object in the image. The authors have shown that the performance is at least as good as human detection and can be obtained with very little training data (200 annotated images).

DeepLabCut, as previously mentioned, is a framework, and despite an excellent documentation [22], it can be challenging to use for a user with little computer skills. The installation process lasts from 10 to 60 minutes and requires a GPU installation to get the most out of the software. Besides, the algorithm requires a lot of computing power. To give an idea, images of 682x540 pixels, analyzed with a last-generation GPU, lead to an analysis speed of 30 frames per second. Without GPU, this time can be multiplied by a factor of 10 or 100 [23].

We see that DeepLabCut is of great interest to precisely find objects in an image. It is particularly aimed at behavioral neuroscience, allowing complex movement tracking (e.g., hand fingers of a mouse). It will not be suitable for users with little computer knowledge interested in more ex-

tensive problems and with little data to process.

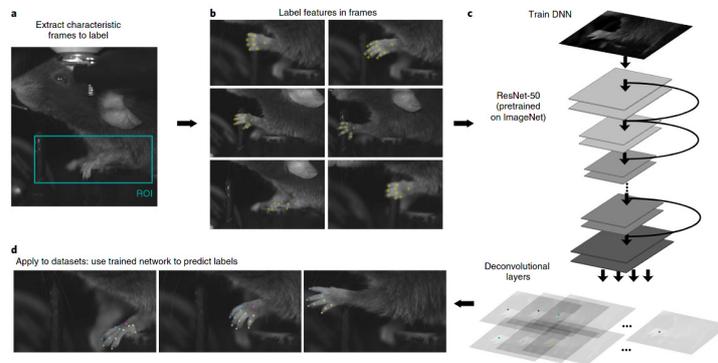


Figure 1.2 – DeepLabCut workflow chart.

idTracker.ai IdTracker.ai [17] is a framework that allows tracking animals with almost perfect accuracy. IdTracker.ai takes advantage of deep learning to carry out the association. In the first step, each object is segmented by applying a threshold. A convolutional network classifies each detected blob as containing a single object or several objects. Another convolutional network finds the identity of each individual throughout the movie.

This system requires enough data to train the network that will recognize each individual. The authors found that robust tracking can be obtained with only thirty isolated images of each individual. Therefore, it is necessary to plan for a minimum of five hundred images for a dozen individuals with a minimum of twenty-five frames per second. A resolution of three hundred pixels per animal is recommended for good tracking accuracy. A limiting factor of idTracker.ai is that it requires a lot of computing time and a lot of RAM. The authors report about twenty minutes for processing a video with eight zebrafish and about six hours for a hundred zebrafish on about two thousand high definition images. Even if a UI is available to help the user, basic computer and programming knowledge is required, and suitable hardware. The use of a GPU is strongly recommended.

This software is suitable for users who want perfect and fully automated tracking from high-quality videos having a powerful computer. A tool is integrated to review and correct the tracking, but the lack of ergonom-

omy makes it sometimes difficult to use.

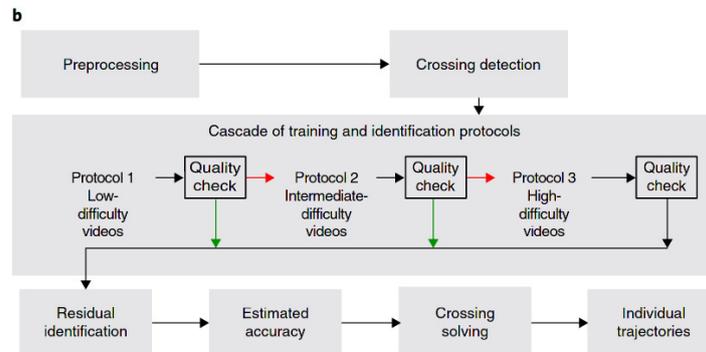


Figure 1.3 – IdTracker.ai workflow chart.

1.4 FastTrack: an original approach

We have previously listed the most used tracking software in different fields of science. We can see that a fast software requiring little computing power, versatile (i.e., that can be applied to any systems with a variable number of objects), easy to install, and open-source is missing. To fill this void, we designed a software called FastTrack [24]. This software contains two distinct parts:

- An interface where standard image analysis procedures are implemented to detect objects, and a tracking algorithm that allows keeping the identity of objects from one image to another, fast and with a low error rate.
- An ergonomic interface where the tracking can be checked and manually corrected if necessary.

FastTrack has a different approach than the software previously mentioned. Instead of exploiting a high computational power to achieve a reliable result without any human intervention, FastTrack implements a simple, very general method. The few resulting tracking errors are left to manual

corrections. In terms of result accuracy, both approaches lead to a quasi-perfect tracking. In terms of speed, human interventions during the post-processing are costly. However, the automatic tracking part is performed much faster, and we noticed that using FastTrack is usually faster. From the images to the trajectories, the duration of the whole process is notably reduced for small projects due to the fast installation and ease of use of FastTrack. Besides, many researchers want to double-check the resulting trajectories to ensure the reliability of the trajectories or get some sense of their objects' dynamics to orient the subsequent analyses, which is performed natively in the FastTrack workflow. This solution has several advantages, the first one being that it does not require any programming knowledge. Any user can perform a perfect analysis in a very short time. Moreover, we will see in the following that the post-processing work can be estimated by an analysis of the geometrical and dynamic parameters of the studied system, which allows the user to know if the software is adapted to his needs. For many of the systems studied, the post-processing is only a quick check. If the number of occlusions is too high, and a perfect tracking accuracy is necessary without having to resort to manual correction, another solution must be considered.

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

Chapter 2

Movies dataset

To demonstrate that FastTrack can analyze movies from various systems, we have compiled a collection of movies named the Two-Dimensional Tracking Dataset (TD^2). This dataset can be downloaded at <https://data.ljup.upmc.fr/datasets/TD2/>. The films either come from data already published in the literature or provided by the authors themselves. All the movies are under a CC-BY-NC-SA license. Each movie is identified by a 3-letter code defining the system (e.g., ACT: active matter, ZFA: zebrafish adult...) and three digits to index films from an identical system. TD^2 currently regroups 41 films, including different types of objects of very different nature and size

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

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

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

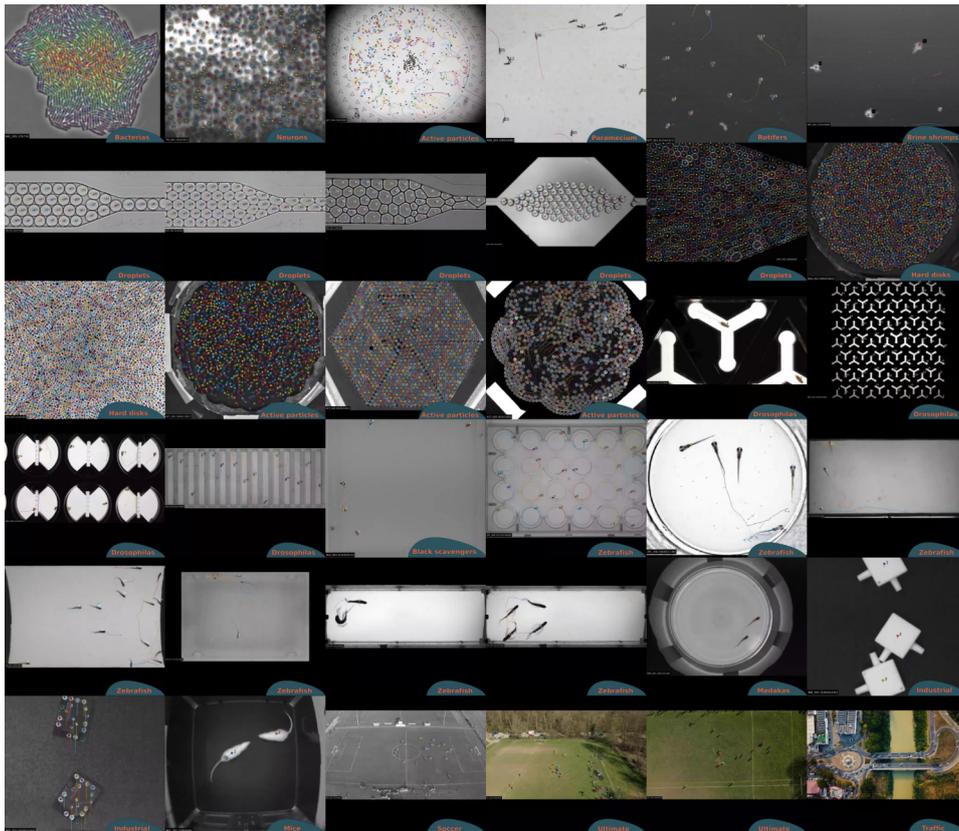


Figure 2.1 – TD^2 dataset Thumbnail of the TD^2 dataset.

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.lip.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 shrimps	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	Unpublished	Yes	Yes
BLS_001	No	2,213	1024x1024	7	Black Scavengers	Black scavengers walking in an arena.	Juan Pablo Busso	Unpublished	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	Unpublished	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	Unpublished	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-laetitia Pontani	16	No	No
DRP_002	No	600	1224x477	Variable	Droplets	Polyhedral droplets evolving in a narrowing microfluidic channel.	Lea-laetitia Pontani	Unpublished	No	No
DRP_003	No	835	1223x434	Variable	Droplets	Polyhedral droplets evolving in a narrowing microfluidic channel.	Lea-laetitia Pontani	Unpublished	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-laetitia Pontani	Unpublished	No	Yes
GRA_001	No	1,001	2040x2040	2043	Hard disks	Vibrated isotropic bidisperse discs.	Olivier Dauchot	Unpublished	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	Unpublished	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	Unpublished	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	Unpublished	Yes	No
PAR_001	No	1,000	1024x1024	Variable	Paramecia	Paramecia swimming in an open field.	Alexis prevost	Unpublished	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	Unpublished	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	Unpublished	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	Unpublished	No	Yes
TRA_001	Yes	1,171	1280x720	Variable	Vehicles	Aerial view of traffic.	No Copyright Footage	Unpublished	No	Yes
ULT_001	Yes	383	1280x720	Variable	Humans	Aerial oblique view of a sequence of Ultimate (flying disk) game.	Paulin Huger	Unpublished	Yes	Yes
ULT_002	Yes	953	1280x720	Variable	Humans and disk	Aerial top view of a half-field Ultimate (flying disk) game.	Paulin Huger	Unpublished	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	Unpublished	Yes	No
ZFA_003	No	2,000	1217x471	3	Adult zebrafish	Three Tupfel long-fin (TL) zebrafish interacting.	Benjamin Gallois	Unpublished	Yes	No
ZFA_004	No	2,000	1212x472	4	Adult zebrafish	Four zebrafish (TL, AB) interacting.	Benjamin Gallois	Unpublished	Yes	No
ZFJ_001	No	200	524x338	14	Juvenile zebrafish	Three weeks-old zebrafish swimming in a flow.	Benjamin Gallois	Unpublished	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	Unpublished	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	Unpublished	Yes	No
ZFL_003	No	1,870	1000x500	3	Larval zebrafish	Three larval zebrafish attracted by a flow with a dye.	Raphaël Candelier	Unpublished	Yes	No
ZFL_004	No	14,830	1024x768	24	Larval zebrafish	Larval zebrafish in a 24-multiwell plate.	Elim Hong, Margherita Zaupa	Unpublished	No	Yes

Chapter 3

Design and implementation

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

Edsger W. Dijkstra

3.1 Tools used

The choice of tools and libraries used in designing software is paramount, 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 license (GPL3), which implies that the language used and the libraries must also be under compatible licenses. The choice of an open-source license is preferable in the case of scientific software [25]. Poor code quality naturally leads to what is called a technical debt. Choosing the "quick and dirty" implementation instead of a time-costly but maintainable solution can hurt the project in the long goal by costing time and effort to run and maintain the code. Open-source can help to solve this problem. First, it is a strong incentive to produce clean code, knowing that it can be read, checked, and used by other people. Moreover, coop-

erative work can help solve bugs faster than a closed-source model. Finally, open-source projects, well documented, can make accessible tools for non-technical scientists that would otherwise have been impossible.

The second criterion is to carefully choose the libraries used, considering the future of the software so that the developers do not have to change libraries if their capabilities prove insufficient as the software evolves. Mature libraries offering long-term support are thus preferred.

In this perspective, FastTrack has been implemented in C++ using the Qt [26] and OpenCV [27] libraries for the graphical interface and image analysis, respectively. Unit tests are performed using the Google Test library.

C++ is a computer language created by Bjarne Stroustrup in 1985 [28]. Offering high performance, it is standardized by the International Organization for Standardization (ISO). It is the language of choice for image analysis applications and the creation of complex graphical user interfaces.

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

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

Google test is a suite for automating unit tests in C++. OpenCV notably uses it. The purpose of unit tests is to verify that each part of the program works as expected. This practice has several advantages: detecting more easily possible errors during the implementation of new features and facilitating software development when it grows in size to avoid any error inclusions. This series of tests are automatically performed on each new commit, see Section 3.3.1 for more information.

3.2 Implementation

FastTrack's operation can be divided into three parts: the objects' detection (detection), the objects association from one image to another (matching), and finally, a manual correction step (post-processing).

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

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

3.2.1 Detection

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

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

- Projection of maximum intensity.
- Projection of minimum intensity.

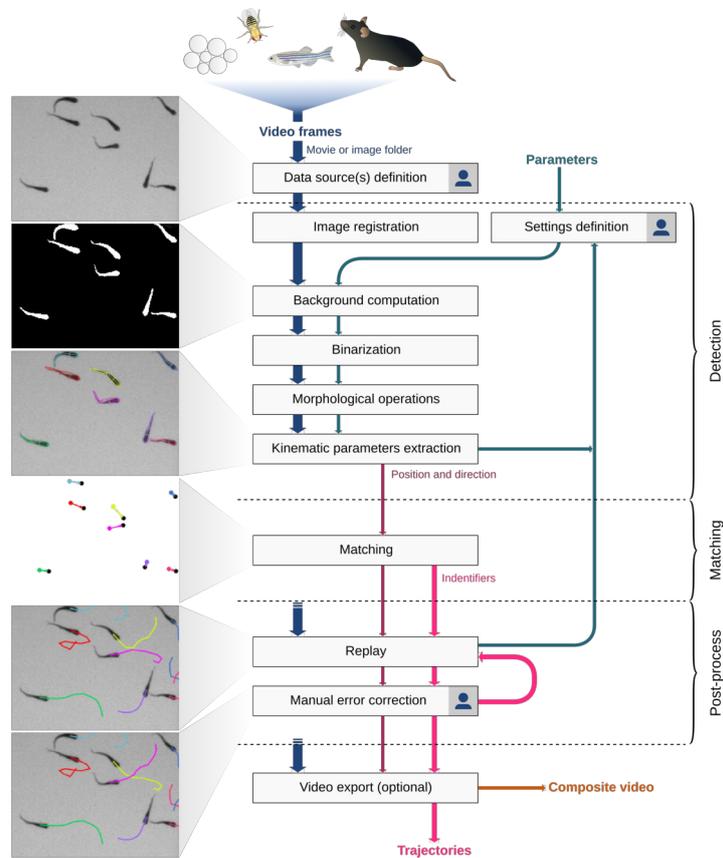


Figure 3.1 – FastTrack workflow chart. The workflow divides into three main parts: detection, matching, and post-processing. A 👤 indicates the few steps that require user input. Sample dataset: *ZFJ_001*.

- Projection of the average intensity.

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 if the objects are darker (resp. lighter) than the background so that the objects disappear and thus obtain the background. The user can make the registration of each image before the projection in order to correct for possible minute camera movements.

Registration The user can choose to register the images. Three methods are proposed in the software. Each method is implemented in a pyramidal way, i.e., the registration is first carried out on a degraded image to roughly correct the displacement. The correction is then refined by increasing the image quality until the original quality is reached. This speeds up the process, as registration is often a relatively time-consuming process.

The first method proposed is phase correlation [29]. It corrects the translational movements between two images using the Fourier theorem in the frequency domain. This method is swift but remains limited to small translational movements only.

The second proposed method is the Enhanced Correlation Coefficient (ECC) [30] method. In FastTrack, it is restricted to correcting translational and rotational movements only. It consists of using the correlation coefficient as a measure to find the best transformation between two images. This method's advantage is that it is relatively fast since this non-linear optimization problem can be solved linearly. It is efficient for noisy images and having photometric distortions.

The third method is a method based on the identification of key points. It allows for correcting movements and deformations (homography). The key points (about 500) are automatically determined on two images thanks to the ORB algorithm [31]. These points are then associated two by two using the hamming distance. The RANSAC algorithm [32] is used to find the best transformation between the two images. This method, more precise, requires a sufficient image quality to be able to discern key points.

Figure 3.2 provides a rough comparison of the performance of the three methods. Using two recordings of the TD^2 dataset, we benchmarked both the accuracy – with the root mean squared difference (RMSD) of pixel intensities between the reference and the corrected image – and the relative computation time. Choosing the right method to obtain the best accuracy depends on each movie's characteristics. However, one can use the rule of thumb that if the objects to track occupy a large fraction of the total area, the best accuracy is more likely to be obtained by using ECC and using the features-based method otherwise. However, as shown in Figure 3.2-C, the ECC method is generally slower by an order of magnitude. Hence, we rec-

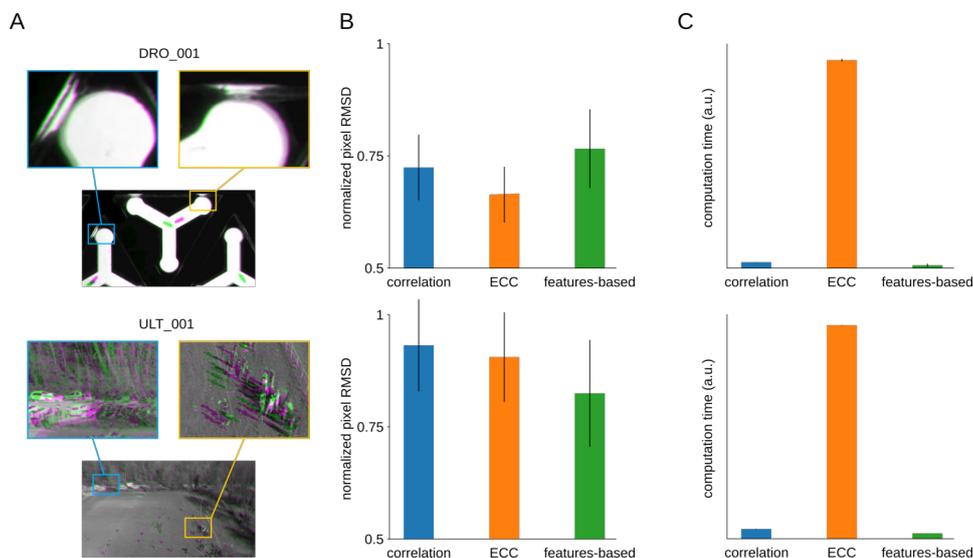


Figure 3.2 – Image registration. Two recordings with severe drift are used for the benchmarking (top: *DRO_001*, bottom: *ULT_001*). **(A)** Comparison of a frame (magenta) with the first frame (green) and magnification of details in the scene. **(B)** The root mean square deviation (RMSD) of pixel intensities 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.

ommend using the features-based method in the general case and long movies.

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

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

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

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.

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 matching step. It is generally with the choice of these quantities that the tracking algorithms can differ to be more specialized for a given type of object. In FastTrack, the parameters extracted are the center of mass, the orientation, the area, and the object's perimeter. These quantities are quickly calculated and general enough to adapt to a wide variety of objects.

To do this, FastTrack calculates the object's equivalent ellipse from the second-order moments of the binary image [33]. This procedure is accelerated by directly using the contour thanks to Green's formula [34]. The object's orientation is given by the ellipse's major axis and is defined in the interval $[0; \pi[$. The direction in the interval $[0; 2\pi[$ is determined by projecting each object's pixel on the major axis of the equivalent ellipse, and calculating the skewness of the distribution of distances of these projected points to the center of mass. The sign of the skewness is a robust indicator of the object's asymmetry along its principal axis. For deformable objects, the previously calculated direction may be different from the direction of

motion. For example, in the case of zebrafish, it bends its body periodically to move. Only the head is directed at the motion. This is why the object is decomposed into two equivalent ellipses. The user can then choose which ellipse best represents the direction of the movement.



Figure 3.3 – Detection Details of the detection phase for one object of the movie *ZFJ_001*. **(A)** raw image. **(B)** binarized image obtained by subtraction the background image and applied a threshold. **(C)** equivalent ellipse of the object. **(D)** two equivalent ellipses, useful for a deformable object.

3.2.2 Matching

The purpose of the association step is to keep the objects' identity from one image to another. To do so, FastTrack uses a method derived from [13], which takes advantage of the fact that each object's position, area, perimeter, and direction changes very little from one image to another.

For each pair of objects (i, j) belonging to two successive images, two costs are calculated: a hard cost that is a threshold set to 1 or $+\infty$, and a soft cost that is a normalization parameter. This terminology is brought from statistical physics, where particles can have soft, long-ranged interactions or hard, binary contacts. The hard cost is defined as follows:

$$\begin{cases} h_{i,j} = 1 & \text{if } r_{i,j} < h_d \\ h_{i,j} = \text{inf} & \text{else} \end{cases} \quad (3.1)$$

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

The soft cost is then defined as follows:

$$c_{i,j} = \frac{\delta r_{i,j}}{s_r} + \frac{\delta \alpha_{i,j}}{s_\alpha} + \frac{\delta A_{i,j}}{s_A} + \frac{\delta P_{i,j}}{s_P} \quad (3.2)$$

where $\delta \alpha_{i,j}$ is the angular difference, $\delta A_{i,j}$ the area difference and $\delta P_{i,j}$ the perimeter difference between objects i and j . To compare these quantities expressed in different dimensions and magnitudes, one need to normalize them. We define the soft normalization coefficients: s_r , 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:

$$C_{i,j} = \begin{cases} c_{i,j} & \text{if } r_{i,j} < h_d \\ \infty & \text{else} \end{cases} \quad (3.3)$$

This cost matrix is, in general, rectangular because the number of objects can vary from one image to the following. A memory parameter can be selected to assign a new identity to an object that disappears on more than the selected number of images. In this case, the row corresponding to this object is removed from the cost matrix and the object cannot be assigned in the subsequent images. We want then to find the best possible matching. This problem is called the rectangular assignment problem and can be solved exactly by using the Hungarian algorithm, see Annexe C. FastTrack uses the Kuhn-Munkres implementation in C++ to solve it.

3.2.3 Automatic tracking parameters

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

It is intuitive to use the standard deviation of the increments of each kinematic parameter. However, some trajectories are needed to estimate

the standard deviations. We set up an iterative, rapidly-converging algorithm to perform this task.

Let us use *ZFJ_001*, a movie with many occlusions and objects of different sizes to illustrate the algorithm's details. For simplicity, let us use only the position, angle, and area as kinematic parameters. There is no gain to expect by adding the perimeter parameter because objects' shapes are very similar. The Figure 3.4-A. shows a snapshot of this movie.

To evaluate the distributions of dr , $d\alpha$, and dA , we start by tracking the movie setting the hard parameters and random soft parameters. The resulting distributions are shown in Figure 3.4-C to E. For kinematic parameters whose differential values can be positive or negative, the distribution is fitted by a Gaussian function, and the soft parameter is set to the standard deviation. For instance, with the angular difference $d\alpha$ the fit reads:

$$f(d\alpha) = \frac{1}{s_\alpha \sqrt{2\pi}} e^{-\frac{d\alpha^2}{2s_\alpha^2}} \quad (3.4)$$

and s_α (orange bar in Figure 3.4-D) is stored as the soft parameter to use during the next iteration. The computation of the soft parameter for the displacement s_r is different since distances can only be positive. Assuming that the displacements along the x and y axes follow two independent Gaussian processes, the resulting displacement follows a χ distribution with 2 degrees of freedom, and the fit reads (see Annexe D for the detailed derivation):

$$f(x) = \frac{x}{\left(\frac{s_r}{\sigma_0}\right)^2} e^{-\frac{1}{2}\left(\frac{x}{\frac{s_r}{\sigma_0}}\right)^2} \quad (3.5)$$

where s_r (orange bar in Figure 3.4-C) is stored as the soft parameter to use for the next iteration and $\sigma_0^2 = 2 - \mu_0^2 = \frac{4-\pi}{2}$.

Once all soft tracking parameters have been derived from the distributions, the software recomputes new trajectories with these updated parameters. This iterative process, depicted in Figure 3.4-B, is run until the tracking parameters converge. In practice, the convergence is very fast, re-

ardless of the initial position in the parameters space. We drew 100 sets of seed parameters from uniform distributions spanning large intervals, and convergence has been attained in very few iterations for all parameters Figure 3.4-F.

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

To characterize the resulting tracking, we computed the number of swaps with respect to the ground-truth:

$$P_{swap} = \frac{N_{swap}}{N_{obj} - n_{ap}} \quad (3.6)$$

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

$$P_{swap} = \frac{N_{swap}}{n(T - 1)} \quad (3.7)$$

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

We computed the converged soft parameters \hat{s}_r , \hat{s}_α and \hat{s}_A for several sampling rates of $\tau > 1$ (Figure 3.4-H to J). We used these parameters to

track the *ZFJ_001* movie at different timescale τ (see Section 4.2) and compute P_{swap} . A comparison between P_{swap} and P_{inc} (see Section 4.2) as a function of τ is shown in Figure 3.4-K. This comparison illustrates that P_{swap} is a noisier measurement of a movie's trackability than P_{inc} and confirms that the iterative algorithm produces trajectories with a number of errors that is close to the statistical limit.

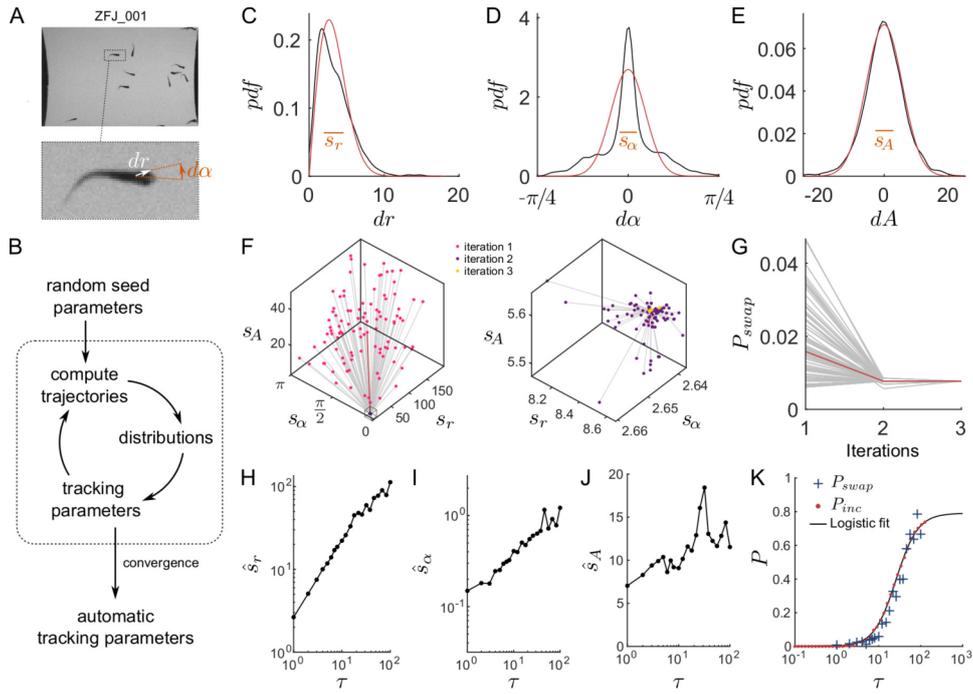


Figure 3.4 – Automatic tracking parameters. (A) Snapshot and blow-up of *ZFJ_001* movie, with definition of \vec{dr} and $d\alpha$ (B) Scheme of the algorithm determining the tracking parameters automatically. (C-E) Distribution of displacements dr (in pixels), angular differences $d\alpha$ (in radians) 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} .

3.2.4 Manual correction

FastTrack integrates a manual tracking correction tool. Once the analysis is completed, the result can be displayed in an ergonomic interface created solely for this purpose. The user can replay the film by superimposing the results of the analysis on the original movie. The user can interactively see each object's parameters. More importantly, the user can also correct tracking errors by deleting objects or exchanging objects' identity. This interface is designed with ergonomics and performance in mind. Keyboard shortcuts and an on-the-fly selection of objects by clicking on the video allow the user to check and correct the trajectories quickly. It is also possible to record a film with the tracking results overlay superimposed.

This manual correction interface makes it possible to shift the workload from the traditional pre-processing of data to the tracking result's post-processing. In the following chapter, we will see how this method can save the user time, the correction time being in general lower than the conception and computational time of system-specific tracking algorithms.

3.2.5 Output

After the tracking, the software generates several files containing the results and the tracking parameters. The result file is named `tracking.txt`, and it contains the raw data of the analysis with one image and one object per line. This format is compatible with the most used analysis software (R, Python, Julia, spreadsheet). Examples in Python and Julia are available in the documentation to get started.

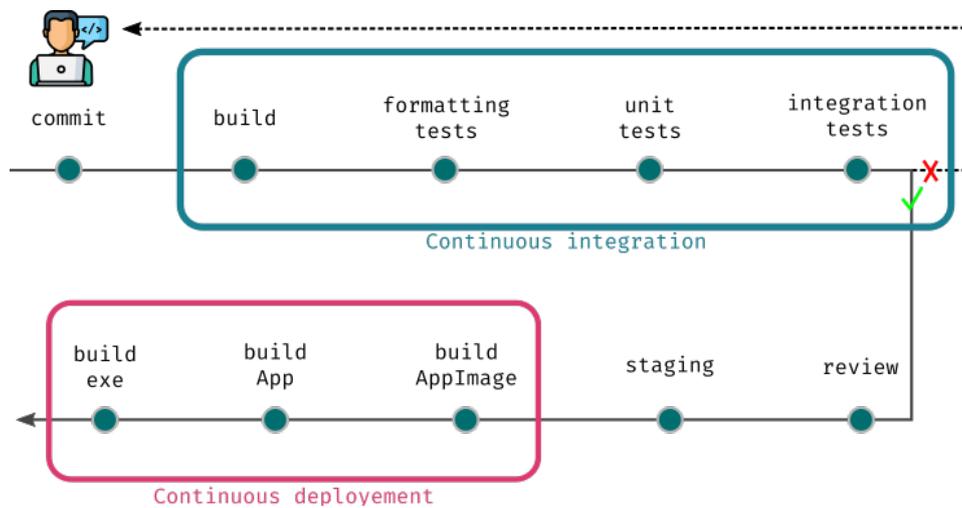


Figure 3.5 – FastTrack CI/CD workflow. The CI/CD workflow is divided into two parts: the CI tasks (blue rectangle) and the CD tasks (red rectangle). The CI must be performed successfully in order to integrate the changes into the project.

3.3 Deployment

3.3.1 Continuous integration delivery

The deployment is one part that should not be overlooked in software design, and two aspects are crucial to consider. From the user's point of view, the software must be easy to install on supported platforms and with fewer bugs as possible. From the maintainer's perspective, the deployment part must be easily achievable and reproducible so that patches and new functionalities can be quickly integrated. From the developer's perspective, the source code's consistency and correctness have to be tested at each change to avoid introducing bugs and facilitate collaboration between developers. With this in mind, FastTrack follows the CI/CD philosophy [35] [36] taking advantage of the recent GitHub Actions system, see Figure 3.5.

Continuous Integration (CI) is a set of practices designed to integrate changes quickly into the project in an automated manner. It is coupled with the automation of unit testing. FastTrack takes advantage of GitHub's

CI/CD system called Actions. With each new (commit¹) or new (pull-request²), a series of tests is automatically triggered. These tests will check the proper functioning of the tracking algorithm and the formatting of the source code. Only the changes that pass the tests can be integrated into the project, which guarantees the reproducibility of the analyses and the source code and documentation consistency.

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

FastTrack natively supports the three most commonly used platforms: Linux systems with an AppImage that supports all distributions, Windows with an installer, MacOS with an App. The latest stable version can be downloaded from the website <http://www.fasttrack.sh>, the nightly build version from <https://github.com/FastTrackOrg/FastTrack/releases>. The procedure to compile the software itself is available in the developer's documentation.

3.3.2 Documentation

FastTrack offers extensive documentation that covers the installation and use of the software. Developer documentation with a documented API and a compilation guide is also available for users wanting to integrate FastTrack in their software or workflow.

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

²Action to request the addition of changes to the project

User User documentation is available at <https://www.fasttrack.sh/UserManual/docs/intro.html>. This documentation is generated from the project, and users can contribute to it at <https://github.com/FastTrackOrg/FastTrack>. It contains all the information needed to use the software and instructional videos to help the user get started with the software.

Developer Developer documentation is available at <https://www.fasttrack.sh/API/index.html>. It is automatically generated by the Doxygen software from the documentation in the FastTrack source code. It contains all the information necessary for developers who want to modify or contribute to FastTrack.

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. *idtrackeraivideoexample* 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.

The benchmark between idtracker.ai and FastTrack was performed on a workstation with an Intel i7-6900K (16 cores), 4.0 GHz CPU, an NVIDIA

GeForce GTX 1060 with 6GB of RAM GPU, 32GB of RAM, and an NVMe SSD of 250GB running Arch Linux. The parameters were set by trials and errors inside the graphical user interface of the two software. The tracking duration was recorded using a script calling the software command-line interface. The average tracking duration and the standard deviation were computed over five runs except for *DRO_002* (2 runs) and *ACT_002* (1 run) due to the very long processing time. Idtracker.ai was evaluated with and without GPU capability except for *100Zebra*, *DRO_002*, and *ACT_002* due to the very long processing time.

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

The accuracy was computed as follows:

$$A = \frac{n_{obj}n_{img} - (2N_{swap} + N_{undetected})}{n_{obj}n_{img}} \quad (4.1)$$

with N_{swap} the number of swaps, $N_{undetected}$ the number of non-detected objects, n_{obj} the number of objects, and n_{img} the number of images. For *100Zebra*, the accuracy was computed only over the 200 first images. All the results are presented in Figure 4.1. As expected, FastTrack is several orders of magnitude faster than idtracker.ai and significantly faster than ToxTrac on all tested videos. That is mainly due to the method used, idtracker.ai using deep learning and ToxTrac cost optimization and the identity preservation algorithm. All software performed exceptionally well in terms of accuracy, except idtracker.ai on *ZFJ_001* probably because the 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 ergonomics in Idtracker.ai.

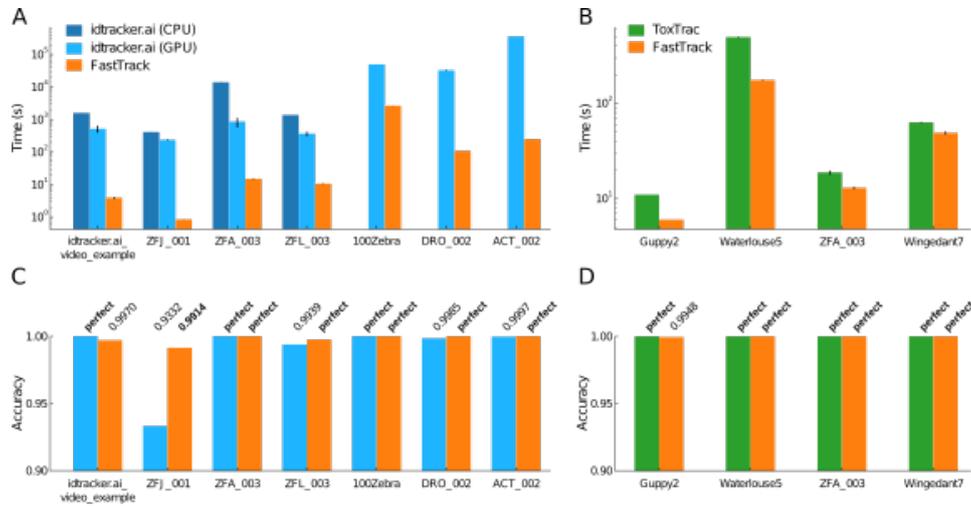


Figure 4.1 – Benchmark of FastTrack, idtracker.ai, and ToxTrac. (A-B) Comparison of the computation time for the tracking of various movies with the same workstation. Whenever possible, CPU and GPU variants of idtracker.ai have been run. Only the first 100 images of *DRO_002* have been used. (C-D) Accuracies 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.

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

4.2 Dataset classification

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

In movies from the academic world, systems are often designed to limit or circumvent these two difficulties. It is common to find movies with a

uniform and constant illumination. Also, quasi-2D confinement and a restricted number of objects in the camera field reduce the number of occlusions.

In TD^2 , 23 movies have an illumination good enough to be analyzed directly with FastTrack. The others had to undergo a specific individual pre-processing before being analyzed. Two movies with too many occlusions were discarded (HXB_001 and ZFL_001) because they could not be analyzed with FastTrack nor by any other software. The remaining 39 films could be analyzed with FastTrack without difficulty. The Kuhn-Munkres algorithm being of complexity $O(n^3)$ the calculation time is generally quite fast. Each film was then manually corrected using the built-in tool to get the ground-truth tracking.

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

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

- the distribution of the displacements,
- the density of objects,
- the geometry of the Voronoï cell
- the degree of motion alignment of the displacements.

To consider the objects' density, we defined the reduced length $\rho = r\sqrt{d}$ where r is the length and d the density. We remark that typically $\rho = 1$ is corresponding to the length between two objects, and $\rho = 0.5$ is the 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:

$$P_{inc} = \int_0^{\infty} R(\rho) p_{inc}(\rho) d\rho \quad (4.2)$$

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

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

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

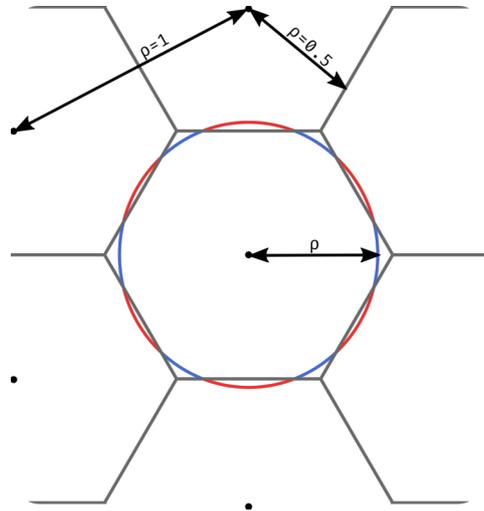


Figure 4.2 – Geometric probability of incursion: Voronoï cells in black and objects black dots with typical ρ length. Circle of radius ρ with portions inside the Voronoï cell in blue and outside in red. The geometric probability for one cell is computed as $p(\rho) = \frac{\Sigma_{out}(\rho)}{\Sigma_{out}(\rho)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

The distribution $R(\rho)$ is shown in Figure 4.3.B for three timescales τ .

A graphical way of calculating P_{inc} is to take the intersection of the areas 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. Therefore we used a fit of the form:

$$P_{inc} = \frac{L}{1 + e^{-k(\log(\tau) - x_0)}} \quad (4.3)$$

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

$$P_{inc} = \frac{L}{1 + \frac{\tau_0}{\tau} k} \quad (4.4)$$

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:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4.5)$$

An exciting outcome of this approach is the ability to determine the optimal 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 Equation 4.4, it reads:

$$\tau_1 = \tau_0(LN_{obj} - 1)^{\frac{1}{k}} \quad (4.6)$$

We computed and ordered the values of τ_1 in Figure 4.3.E for the whole dataset. It appears that three quarters (30) of the movies are oversampled. Any difficulty in the tracking should not be expected concerning incursions. On the other hand, nine movies are undersampled. These recordings were already known to be difficult to track, three of them (*ACT_003*, *ACT_004*, and *GRA_003*) have required specific algorithms for analysis, and two (*BAC_001*, *ZFA_001*) required dedicated software.

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

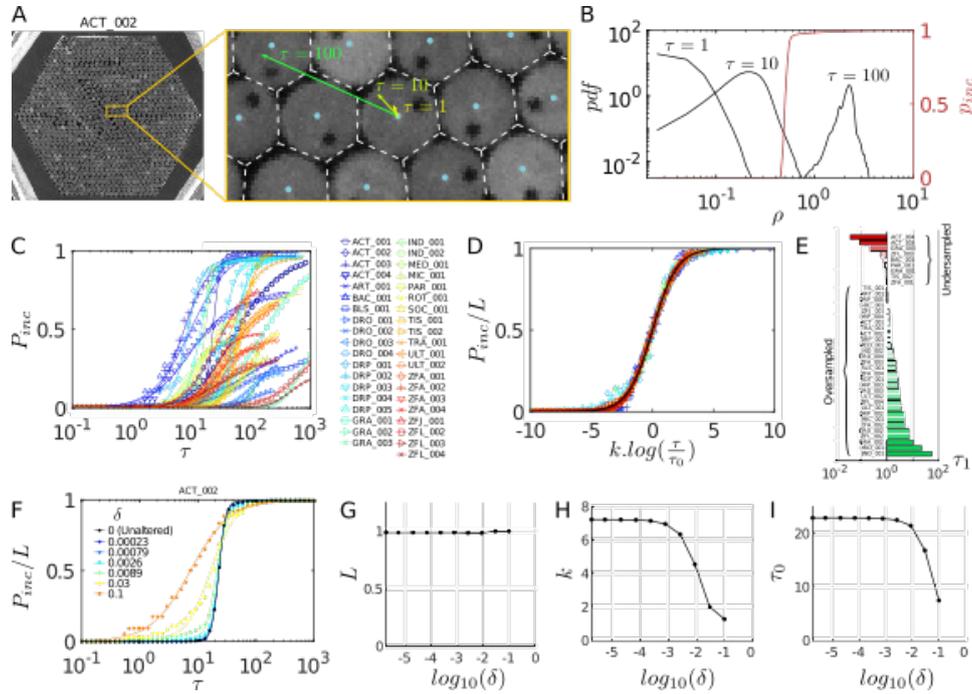


Figure 4.3 – Characterization of the TD² dataset. (A) Illustration of the dynamics at various timescales in *ACT_002*. The Voronoi cells (dashed white) and the displacements of a particle at $\tau = 1, 10$ and 100 are overlaid. (B) Geometric probability of incursion p_{inc} (red) and distribution of 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 $k \cdot \log(\frac{\tau}{\tau_0})$ 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. (4.6), 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*

4.3 Parameters optimization

One may also want to determine the optimal tracking parameters, i.e., with a P_{swap} as close to 0 as possible. Provided that the ground-truth is known for at least one movie, for example, by a careful manual post-

processing. It is possible to leverage FastTrack’s speed to explore the parameters space and minimize P_{swap} . The optimized parameters found that way can be used to track other similar movies with a minimal error rate. The workflow of the method is depicted in Figure 4.4-A. As the exploration of the whole parameters space requires to perform at least thousands of trackings, such an approach is only made possible by the command-line interface (CLI) and the speed of execution of FastTrack.

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 considered 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 tessellation) 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.

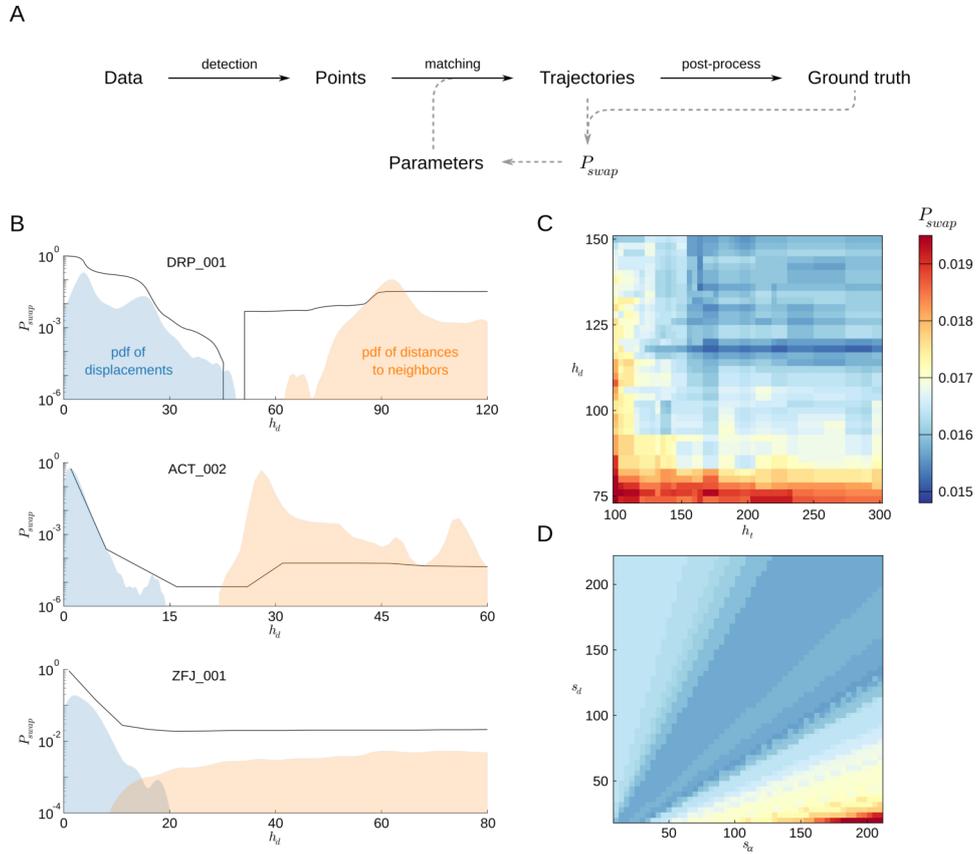


Figure 4.4 – Optimization of tracking parameters based on P_{swap} . (A) Scheme of the optimization workflow: on top of the detection/matching/post-process flow chart, the ground truth is used to compute P_{swap} and create a feedback loop on the tracking parameters. (B) P_{swap} (black) as a function 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 $h_t = 90$.

Chapter 5

Conclusion and perspective

In these chapters, we saw how we implemented a versatile and easy to use tracking software using open-source tools. Taking advantage of the GitHub Actions system, we automated the testing and the deployment of the software, increasing confidence, and promoting collaboration. We have shown that FastTrack can compete with state-of-the-art tracking software for many use cases. At the same time, we compiled a dataset of movies, allowing us to benchmark tracking software over a wide variety of systems. We classified the dataset based on the probability of incursion and, doing so, defined a criterion to determine the optimal framerate of acquisition. We have finally shown how to determine the best set of tracking parameters by leveraging FastTrack's full capabilities.

FastTrack's original approach, shifting the workload on the post-processing phase while keeping the pre-processing as light as possible, allows the use of FastTrack on a wide variety of systems. The post-processing phase, mainly a swift checking of the tracking and small corrections, can be done directly inside the software in an interactive and ergonomic environment. FastTrack allows users to track movies quickly without any computer knowledge.

FastTrack's approach does not prevent human inputs, mainly in the post-processing phase, to obtain a perfect tracking accuracy. It will be without inconvenience for many users who will need a human checking in any

case. However, users who want a perfect tracking accuracy without human input will have to turn onto other tracking software.

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

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

Appendices

Appendix A

FastTrack user interface preview



Figure A.1 – FastTrack’s user interface FastTrack’s main window with ZFJ_001 movie opened.

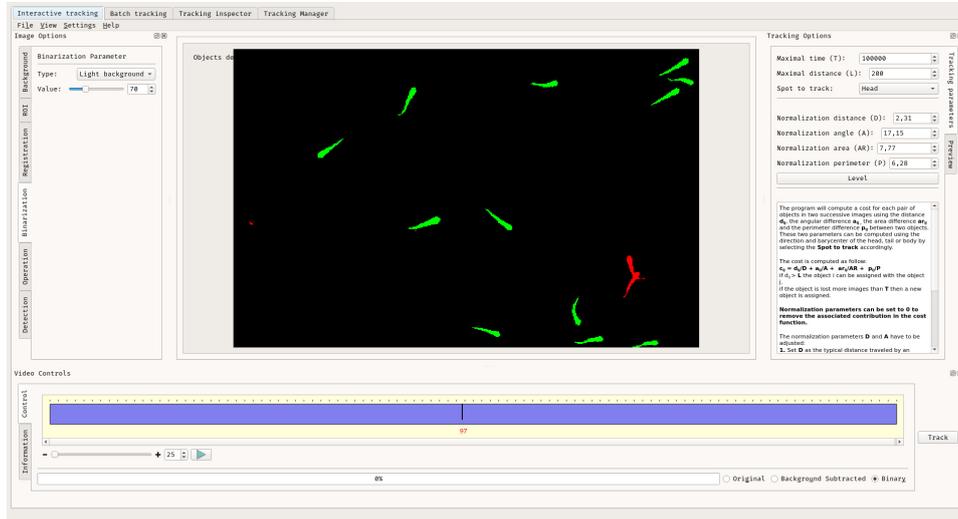


Figure A.2 – FastTrack’s user interface FastTrack’s main window with *ZFJ_001* movie opened and parameters setup for the tracking.

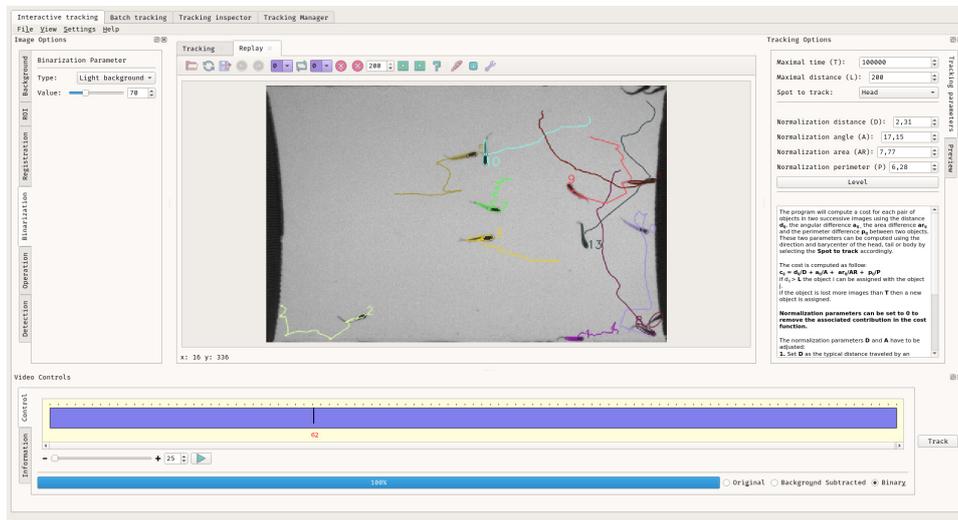


Figure A.3 – FastTrack’s user interface FastTrack’s tracking review with *ZFJ_001* opened after the tracking. The user can check and correct tracking errors.

Appendix B

Voronoi diagram

B.1 Definition

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

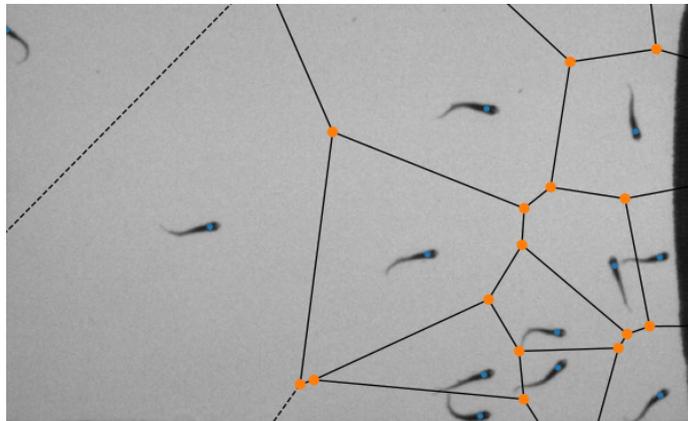


Figure B.1 – Exemple of a Voronoi diagram computed with one image of *ZFJ_001*. Voronoi vertices represented with orange points, seed points with blue points, finite Voronoi ridges with black lines, and infinite Voronoi ridges with dashed-black lines.

B.2 Construction

In general position ¹, the dual graph of the Voronoï diagram is the Delaunay triangulation. The Delaunay triangulation is a triangulation where every circumcircle is an empty circle. The circumcenters of Delaunay triangles are the vertices of the Voronoï diagram.

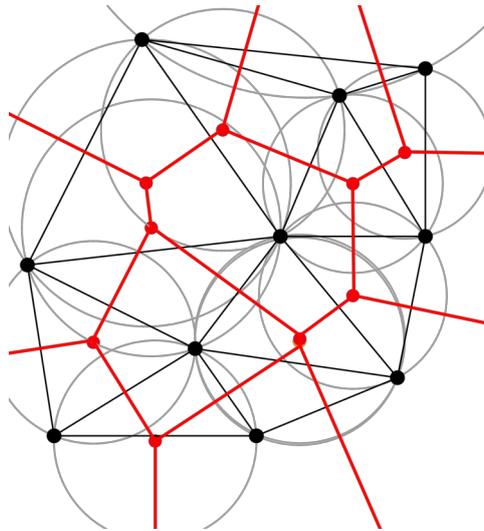


Figure B.2 – Delaunay triangulation and Voronoï diagram. Delaunay triangulation in black, circumcircles in grey and Voronoï diagram in red.

¹An arrangement of points with no three collinear.

Appendix C

Hungarian algorithm

C.1 Definition

The Hungarian algorithm is a combinatorial optimization problem that solves the so-called assignment problem in polynomial time [37]. Since 1957, it has been known as the Kuhn–Munkres algorithm [38] after that James Munkres reviewed it as strongly polynomial. First $O(n^4)$, several implementations exist with a complexity of $O(n^3)$ [39–41].

C.2 Description

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

¹<http://www.hungarianalgorithm.com>

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

Step 1: subtract the row minimum from each row:

0	13	78	45	(-14)
21	26	33	0	(-17)
0	32	54	57	(-10)
56	0	51	57	(-32)

Step 2: subtract the column minimum from each row:

0	13	45	45
21	26	0	0
0	32	21	57
56	0	18	57
(-0)	(-0)	(-33)	(-0)

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

0	13	45	45	
21	26	0	0	<i>x</i>
0	32	21	57	
56	0	18	57	<i>x</i>
	<i>x</i>			

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:

0	0	32	32
34	26	0	0
0	19	8	44
69	0	18	57

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

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

Appendix D

Fitting of the distribution of displacement

The displacement is the square root of a sum of squares of two independent gaussian variables ($\Delta l = \sqrt{(\Delta x)^2 + (\Delta y)^2}$), thus the displacement follows a χ distribution with 2 degree of freedom. The standardized χ distribution with 2 degree of freedom reads:

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

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:

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

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

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

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

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

Part II

Assaying chemical preference of the young zebrafish

Chapter 6

Introduction

6.1 The chemical perception

The chemical senses are one of the oldest sensory systems [42]. They are the most used sensory modality and observed in a wide range of taxa, from unicellular [43] to mammalian. Some features and basic principles are highly conserved across phyla [44,45], and mediate several behaviors like predator avoidance, food-finding, and mating necessary to the survival of any species.

Fish are immersed in a complex chemical environment at any time and have evolved a refined sensory system to perceive and interpret these chemical stimuli. For fish, chemical perception is mediated by three senses: olfaction (smell), gustation (taste), and a common chemical sense. Unlike terrestrial species, where substances perceived by smell and taste differ by the medium of transport of the molecules, fish taste and smell through the same medium: water. The solubility of compounds in water determines the type of compounds that can be transported and perceived. Therefore, chemical perception is non-directional and persistent. The distance traveled and the perceived concentration depend on the diffusion and convection of the medium, determining the perception threshold and the compound's residence time in the environment. The chemical perception is extremely specific, being contained in the molecular structures and the

complex mixture of chemicals.

Mechanisms of perception have been well studied for diverse fish species [46], but highly complex directed behaviors like homing migration or food-finding are still poorly understood [47]. For example, fish can find food in complex environments like turbid and turbulent waters, where the perception is fragmented. Deciphering these mechanisms and their associated neural mechanisms will significantly advance the comprehension of the animal kingdom's most used sensory modality.

One fish species, the zebrafish, is an emerging model for studying goal-driven behaviors. At the larval stage (6 days post-fertilization), the animal is transparent, and it is possible to observe the brain activity with cellular resolution using light-sheet microscopy [48]. The development of virtual reality assays makes it technically possible to associate some neuronal networks' activity to the observed behavior. Using this technique; it was possible to gain insights into several behaviors, such as prey capture [49], optomotor response [50], phototaxis [51], rheotaxis [52], and thigmotaxis [53].

The development and functioning of the fish sensory organs, particularly in the zebrafish, have been well characterized. However, there are few behavioral studies on chemical perception and chemically-oriented navigation. Several milestones have to be achieved before using virtual reality assays to study chemically-driven navigation. One needs to find and characterize products that elicit robust and attractive behaviors. The space of possibilities is vast, and completing this task will necessitate a high-throughput setup to explore combinations of products, concentrations, and fish ages. In a second time, when a product that elicits a robust and attractive response will be characterized, studying realistic chemically-driven navigation will require a setup capable of reproducing turbulent flows where the fish is immersed and submitted to a complex chemical environment with fragmented perception.

In the next sections, we will present the fish's sensory organs and review experimental setups used in the literature to characterize the zebrafish's chemical perception at the larval and adult stage. Then we will present two experimental setups that we built to study the chemical per-

ception of zebrafish. Finally, we will present and discuss some results that we obtain using these setups.

6.1.1 Olfaction

The olfactory organ of the fish, see Figure 6.1, consists of two structures located in the animal's snout [46]. Each structure consists of a cavity called the olfactory chamber connected to the outside by an entrance and an exit nostril. The inside of the olfactory chamber is lined with the olfactory rosette consisting of two rows of olfactory lamellae. The olfactory epithelium, where the olfactory receptors are located, is placed on these lamellae. The olfactory organ's exact organization and position can vary depending on the fish species, for example, with the addition of a ventilation cavity as an extension of the olfactory cavity.

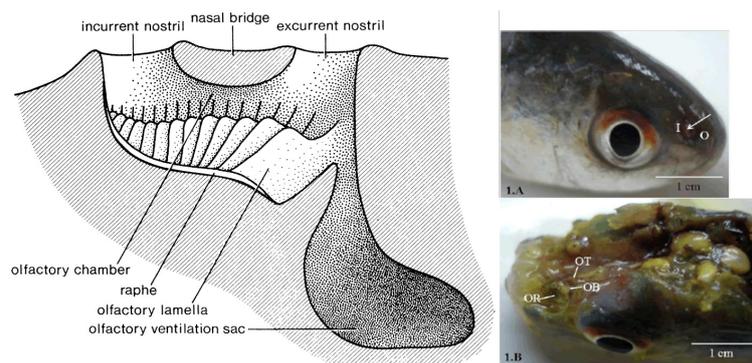


Figure 6.1 – Olfactory system, reproduced from [46]. The olfactory organ of *Labeo bata* reproduced from [54], (I) is the inlet and (O) the outlet channel, (OR) olfactory epithelium, (OT) olfactory tract and (OB) olfactory bulb.

The olfactory epithelium has a $100\mu m$ thick pseudostratified columnar structure [46]. It can be separated into a sensory and a non-sensory epithelium. The sensory epithelium consists of three types of cells: receptor, supporting, and basal cells; the non-sensory epithelium of goblet cells and non-sensory ciliated cells. There are five receptor cells implicated in the olfactory perception: ciliated cells, microvillous cells, crypt cells [55, 56], kappe cells [57], and pear-shaped cells [58]. They express olfactory receptors of the OR, V1R, V2R, and TAAR families. Receptor cells have various sizes, shapes, and distribution inside the epithelium, see Figure 6.2.

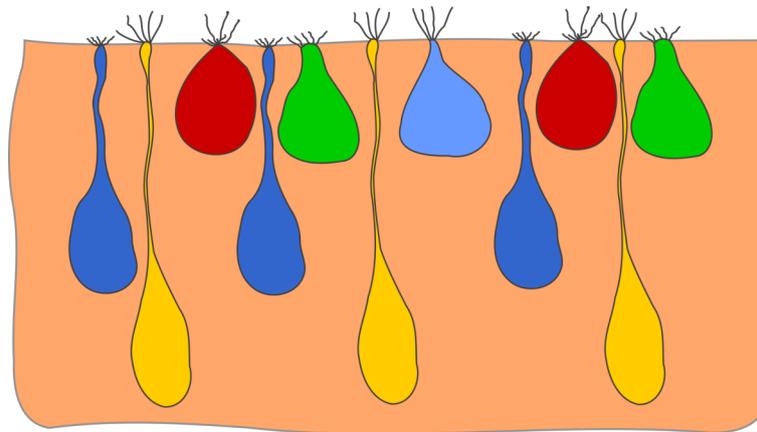


Figure 6.2 – Schematic representation of the olfactory epithelium. Ciliated neuron in yellow with round somata, slender dendrite, and cilia. Microvillous neuron in dark blue with microvilli at the surface. Crypt neuron in red with ovoid shape, microvilli, and cilia. Kappe neuron in green with microvilli.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

Receptor cells project directly into the olfactory bulb located in the brain, in turn sending signals to the telencephalon and diencephalon [59]. The olfactory bulb in the teleost is a structure of four concentric layers: olfactory nerve layer (ONL), glomerular layer (GL), mitral cell layer (MCL), and internal cell layer (ICL). The olfactory information is transmitted by the receptor cells to the olfactory bulb [60] then in the forebrain [61] as a topographical odor map. The olfactory bulb's neuronal connections have been particularly studied in the zebrafish [62, 63], the olfactory bulb comprised approximately 20 000 neurons [64] and 140 glomeruli [65]. Each receptor cell expresses only one type of olfactory receptor [66–69] except in a subpopulation of olfactory sensory neurons [69]. Cells expressing the same receptor are projecting into the same olfactory bulb glomeruli [70]. Glomeruli responding to similar odorants are grouped into domains within the olfactory bulb, forming chemotopic maps. Odorants can activate glomeruli outside their domain, leading to a fragmented map inside the olfactory bulb [71]. Moreover, the odor encoding is hierarchized with first-order features encoded by large domains and second-order features by local activity patterns within the domain [72, 73].

The olfactory bulb projects into two higher brain structures, the telencephalon (Dp and Vv) and the diencephalon (habenula, posterior tuber-

cle, and hypothalamus). The neuronal activity evoked by olfactory cues in these areas is currently poorly understood [63].

In the zebrafish [74,75], the olfactory organ develops from the olfactory placodes at the 6-10 somites stage (about 15 hours post-fertilization) of the embryonic development. The olfactory cavity begins to appear at the 28-30 somites stage (31 hours post-fertilization). Approximately 50 hours post-fertilization, the olfactory epithelium and the receptor cells appear. When the embryo exits from the chorion at 4 days post-fertilization, the olfactory organ continues its morphological development, but the cytological organization remains largely unchanged. At 40 days post-fertilization, the bridge between the entrance nostril and the exit nostril is completely formed, separating the currents going out and coming in from the olfactory cavity. The addition of lamellae to the olfactory rosette continues throughout the life of the zebrafish.

6.1.2 Gustation

The gustatory organ of fish is composed of taste buds that are not localized in a single location but rather spread all over the body surface and directly contacting chemical substances. Taste bud histology has been studied for different fish [76-80], and they usually have an elongated and ovoid shape, see Figure 6.3. They sit on a small dermal papilla and extend throughout the epidermis' thickness protruding from the surface. The taste bud is constituted of a sensory (dark cells with microvilli and light cells with one large microvillus) and a non-sensory (Merkel-like basal cells) epithelium. The apical ending of the sensory cells that protrude from the epithelium is called the receptor field and is covered with a mucous cap. The number of sensory cells in a taste bud varies considerably depending on the fish species, 2 200 on the zebrafish up to 1 000 000 in large *Ictalurus nebulosus* [81].

Taste buds are distributed all over the fish's body, especially in the mouth, on the lips, and the skin. Their distribution and concentration vary according to the species. Three different cranial nerves innervate them: facial (VII), glossopharyngeal (IX), and vagal (X). The facial nerve transmits

information from the extra-oral taste buds; the glossopharyngeal nerve transmits information from inside the oral cavity; the vagal nerve transmits information from inside the oropharyngeal cavity. The taste system is anatomically divided into two distinct parts: nerves IX and X projecting into the brain's vagal lobe and nerve IV into the facial lobe. Connections to higher areas of the brain differ slightly from one species to another. It has been shown in *Ictalurus nebulosus* [82] that these two systems have distinct roles in fish feeding behavior.

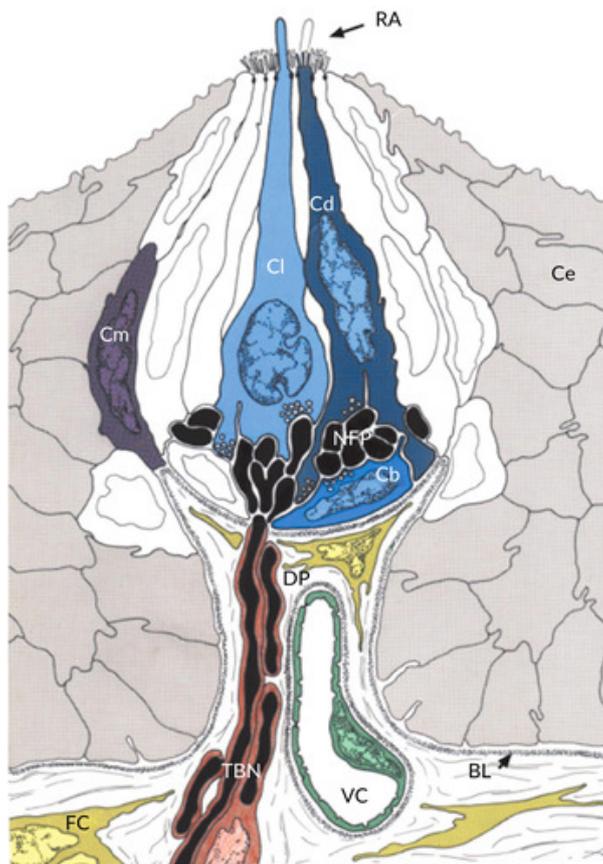


Figure 6.3 – Schematic drawing of a typical taste bud of teleosts from [83]. Dark cells (Cd), light cells (Cl) and Merkel-like basal cells (Cb). Marginal cells (Cm). Ce epithelial cells. Dermal papilla (DP). (TBN) taste bud nerve. (BL) basal lamina. (RA) receptor area. (VC) capillary vessel.

In zebrafish [84], the taste buds (approximately 2 200) are located on the lips, in the oropharyngeal cavity on the barbels, and on the head's ventral and dorsal side. Each taste bud contains 20 to 23 cells. Projections

of the zebrafish gustatory system have been studied in detail [85] and form a complex network that can be summarized graphically see Figure 6.4.

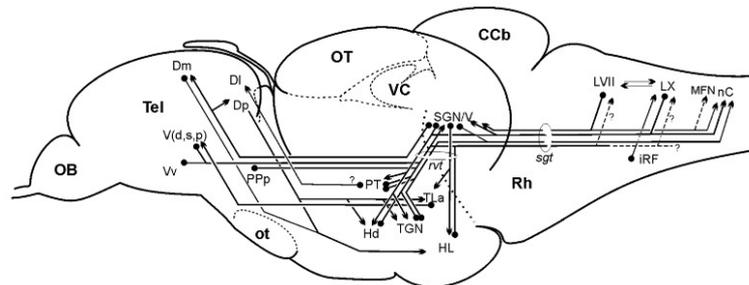


Figure 6.4 – Gustatory system of the zebrafish. Neuronal connections of the fish taste system reproduced from [85].

The development of the gustatory organ has been studied in the zebrafish [83]. The first taste buds appear at 3 or 4 days post-fertilization and are located on the lips and the gill arches. The taste buds in the mouth and oropharyngeal cavity appear 4 to 5 days post-fertilization. The taste buds on the head do not appear until 12 days post-fertilization, and it is not until the juvenile stage (30 to 40 days post-fertilization) that the barbels appear. Note that the appearance of the taste buds coincides with the appearance of feeding in the larvae.

6.1.3 Common chemical sense

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

In the zebrafish [86], SCCs have been described as a set of 2-7 villi of 0.5 to 1 μm length emerging from the cell body at embryo and larval stage. In adults, SCCs possess a single villus of 3 μm length.

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

6.2 Behavioral studies

6.2.1 Behavior

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

A well known and impressive behavior encountered in many fish species is the homing migration. A typical example is salmon that perform three migratory phases throughout their life. One of them, the upstream migration from the ocean to their home stream, has been shown to rely on an olfaction imprinting [87,88]. Little is known about the imprinting mechanism, but experiments suggest that it relies on a mixture of odors perceived during the juvenile stage in the fish's home stream.

Feeding is one of the most important behaviors. It relies on several senses for food detection and selection [89]. A stereotyped behavioral sequence was shown to exist in many species [90] consisting in a step of arousal mainly mediated by olfaction [91], then a step of localization of the food mediated by chemical and visual cues. The last step of ingestion is triggered primarily by the gustation [90]. The impact of each sensory modality varies significantly with the species. For example, the yellow bullhead has the entire feeding sequence mediated by taste, whereas ictalurid catfish prey detection was abolished when olfaction was blocked. The chemical substances that attract fish depend on the species [90], and response to a mixture is higher than isolated compounds in general.

Olfaction [92] as well as the gustatory system [93] has been shown to play an essential role in reproduction. Non-anosmic males exposed to water taken from a tank with a gravid female developed courtship behaviors, except for some species like the three-spined stickleback where the gustation can replace the olfaction. Complete courtship repertoire necessitated the presence of other sensory cues.

Fright reaction occurred when a fish perceived an alarm substance se-

creted by a conspecific. This reaction differs between species and involves seeking cover, rapid swimming, or freezing. It is accepted to be mediated by olfaction [94–96], but other sensory cues are not ruled out.

Most of the works done on the zebrafish chemically induced behaviors focus on developing the model for pharmacological safety screening [97], drugs addiction [98], and ecology [99], enabling a low cost and genetically manipulable model. Behavioral studies of the chemical perception of zebrafish, adults, or at the larval stage have been done through various experimental assays that will be presented in the following sections.

6.2.2 Conditioned place preference

The conditional place preference (CPP) experiment is a type of Pavlovian conditioning. Pavlovian conditioning consists of associating a conditioned stimulus (generally neutral) with an unconditioned stimulus. After learning, the animal exhibits a conditioned response to the conditioned stimulus when presented alone. The most classic example is associating a bell's sound (conditioned stimulus) to the release of a food smell (unconditioned stimulus). After learning, the animal can respond to the bell's sound alone, as demonstrated by Ivan Pavlov on dogs [100].

This approach was applied to test the response to various chemical stimuli in adult zebrafish [101]. The experiment follows a classical 3-step design. The first step is to evaluate the fish's base-line preference. The animal is placed in an aquarium with two or three distinct areas differentiate by walls' pattern and color, see Figure 6.5. The fish is tested to find out which side it naturally prefers. In this experiment, the distinctive walls' pattern and color play the role of the conditioned stimulus. The second step is the conditioning phase. The fish is restrained to its least preferred area, and the substance to be tested injected into the water (unconditioned stimuli). The last step consists of repeating the first step to assess the change in preference of the animal.

Several chemical substances have been tested using this method [102–104]. Notably, a strong and robust cocaine-induced CPP response in WT ze-

brafish was shown [105], with 85% of the fish changing preference to a cocaine concentration of 10mg.L^{-1} and lower and higher concentrations resulting in a lower response. A positive response of adult zebrafish to a single ethanol exposure was shown [106] in a similar experimental setup. It should be noted that this is also the first study to use an automated tracking system to calculate animal preference. Zebrafish showed a positive response for D-amphetamine [107, 108], salvinorin A [109], cocaine [109], spiradoline [109], nicotine [110] and ethanol [110].

We see that CPP has been used extensively to study the response to chemical stimuli in zebrafish. There is a strong emphasis on products that cause addictive pathologies in humans. Nevertheless, this protocol has several limitations, the most important being that it involves several systems of perception as well as memory. During the conditioning phase, the learning is based on the visual perception of the environment (pattern on the aquarium walls), the chemical perception of the tested compound, the association of the two stimuli coming from different sensory organs, and the memorization of these perceptions. Secondly, the time window to perform the experiment (minimum two days) and the difficulty to automate it is a hindrance to use the CPP to study the effect of many chemicals in a high-throughput manner. Furthermore, this protocol is too complex to be used with larval zebrafish and coupled with neuronal imaging.

6.2.3 Multi well-plate

A widely used experimental apparatus to assess chemical compounds' effect on zebrafish larvae and embryos is the well-plate device [111]. One or more larvae is placed in each well in a bath of a chemical. Larvae are then recorded swimming in the chemical compound, and the kinematic parameters of the animal are extracted. In the case of embryos, development is monitored after exposure. The advantage of this technique is that it requires only a single experimental apparatus. It quickly produces a large amount of data with up to 48 wells per plate. Software already exist to extract automatically relevant behavior parameters from video recordings [112].



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.

With this kind of assay, many chemical compounds have been tested [113–115], as well as seizure liability [116], and several behaviors [117–120].

The well-plate device allows for an easy and automatic high-throughput screening of chemicals. Turnkey commercial solutions like the ZebraBox from ViewPoint exist, and custom setups are relatively easy to build. However, this system suffers limitations like the fact that one can not assess the fish preference. Precisely controlled exposure, or repeated exposure through cycles of exposition/flushing, are not available. Therefore this system is not adapted to investigate fish's chemical preference and chemical-driven navigation.

6.2.4 Direct introduction

Some authors have tried to quantify chemically induced behavior by introducing a chemical compound directly into the tank and looking at the percentage of time spent close to the source. Notably, an attraction concentration-dependent to adenosine and ATP for adult zebrafish [58]



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

and to GCDA and nicotine for zebrafish larval [121] was shown. A strong aversion to cadaverine, an odor associated with decomposing bodies, was shown using a tank with a single compartment or a tank with two compartments and an intermediate zone where the fish can change compartment [122], see Figure 6.7.

Very easy to implement, these types of experimental devices lack control in the concentration perceived by the animal. Diffusion and advection are neglected in the experiment, and the concentration is poorly known and not reproducible. Moreover, these setups exclude the realization of long experiments due to the homogenization of the product.

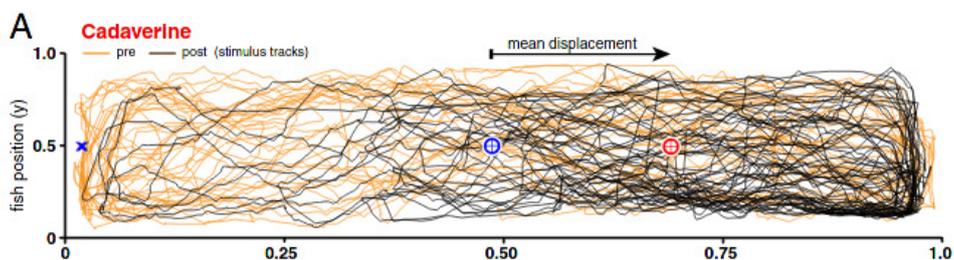


Figure 6.7 – Diffusion setups from [122]. A. One channel diffusion setup, blue cross: chemical introduction point.

6.2.5 Flow

Another type of device that allows the product's concentration inside the tank to be quickly changed was used on adult zebrafish [123], see Figure 6.8. Like in the multi-well experiment, chemically induced behavior changes were monitored by the animal's kinematic parameters.

Several food odors were shown to produce a significant increase in speed and number of bursts; social odors from conspecific produced a similar response; alert odors result in a dive to the bottom of the tank and an increase in frozen time; decomposition odors result in more turns. The critical points noted with this device is the inter-and intra-experimental variability. The authors showed that less than a third of the odors used in the study produce reproducible results between trials of the same individual. Some odors such as cadaverine, blood, skin, and food odors resulted in inconsistent responses for the same individual. Most odors produce poorly reproducible results for different fish.

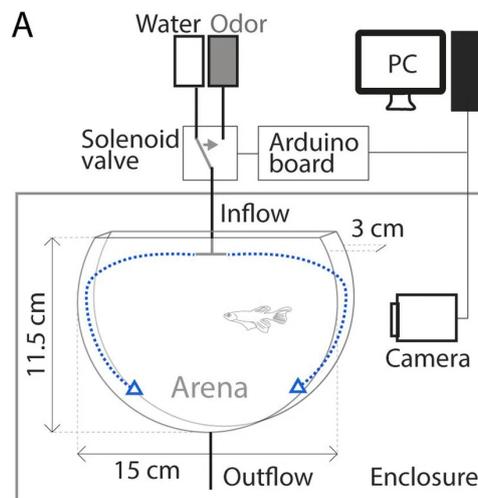


Figure 6.8 – Flow setup from [123].

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

A more controlled setup to study chemical preference in fish is the underflow device. The first mention of this type of device dates back to 2013 [124]. In this setup, the tank is separated into two distinct compartments using a laminar flow, see Figure 6.9. 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 crosses, and the animal's kinematic parameters are extracted from video recordings to assess the fish's preference.

Several psychoactive substances have been tested on adult zebrafish [125, 126] and showed attraction by diazepam, fluoxetine, risperidone, and buspirone; neutral response to ethanol and clonazepam; an aversion to acid pH, two food odor extracts, and conditioned water took from a tank with chemically and physically stressed fish.

This setup has several advantages. The product's concentration is perfectly known because the diffusion and advection are mitigated and controlled by the flow. The fish's preference can be directly measured as the fish can choose freely to go inside or outside the product. Long experiments can be performed with this setup, and product delivery precisely controlled in time. However, some disadvantages remain, like the absence of a standardized or turnkey setup and the volume of water and chemicals required that can be high.

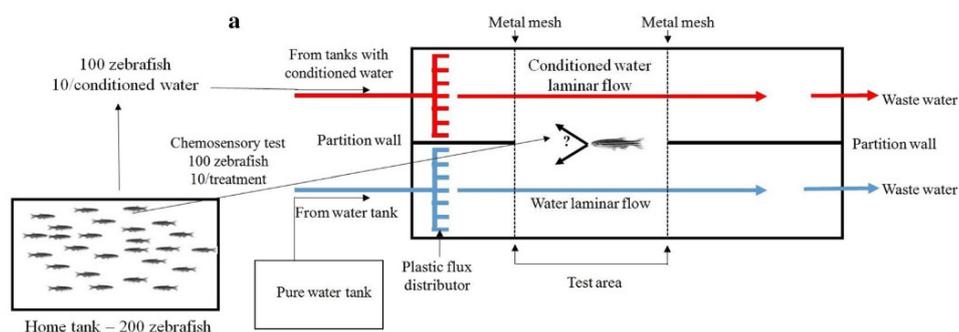


Figure 6.9 – Flow setups from [126] The flow is separating the tank in two lines (left-right).

From this overview of the scientific literature, we see that the study of

chemical perception and behavioral response to chemical stimuli is not standardized. Direct comparisons between studies carried out in various independent laboratories are not easily possible. In this context, we have developed Dual, an open-source, easy to replicate, low cost, do it yourself, and scalable experimental setup. Using the underflow principle, Dual allows studying chemical preferences in larvae and juvenile zebrafish in a standardized, high-throughput, and comparable way.

Chapter 7

Experimental setups

To study the young zebrafish chemical perception, we build two complementary setups: one to screen products, the other to recreate more natural flows, and studies the chemically-driven navigation.

7.1 Dual

The design and implementation of experimental setups that are high-throughput, bias-free, and scalable are essential to behavior characterization. We present Dual, a high-throughput experimental setup that is easy to build, scalable, and costs less than 2 000 euros.

7.1.1 Overview

Dual is an underflow system built on the same principle as [124]. It consists of creating two compartments in a tank through a laminar flow without any physical separation. As we have seen previously, this system allows a rigorous knowledge of the compound's concentration to which the animal is subjected. Diffusion and advection due to the water movements caused by the fish are avoided. The interface between the two compartments is well defined and self-healing when disturbed, with a charac-

teristic time depending on the flow rate.

To create the laminar flow, Dual uses a system of four syringes, coupled two by two: when two syringes push, two syringes pull, see Figure 7.2. The syringes are connected to the inputs and outputs of a millifluidic chip and generate a laminar flow with constant volume by having two syringes injecting at one side and two aspirating at the other side. The fish is placed inside the millifluidic chip, see Figure 7.6. Thanks to a computer-controlled manifold composed of six microvalves, each syringe can be filled independently. One can then build an experimental protocol by stacking several filling and injection cycles and choosing what to fill the syringes with.

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

Dual is a custom-built system using several components stemming from the makers' community (Openbuilds, Markerbeam, Arduino), which allows great flexibility in the conception and fast iteration to adapt the project if necessary. All the components, blueprints, and other CAD files needed to build and assemble Dual are freely available. Dual can be built at a low-cost (see the bill of materials Appendix E) without prior knowledge of mechanics or electronics. The tools necessary for realizing Dual can be found in a FabLab and necessitate little formation.

7.1.2 Construction

For the construction, Dual can be separated into five main parts. The mechanical system comprised the static structures, the motor, and moving parts. The millifluidic system is constituted of the millifluidic chip, the microvalves, the syringes, and connecting tubes. A camera, lens, infrared

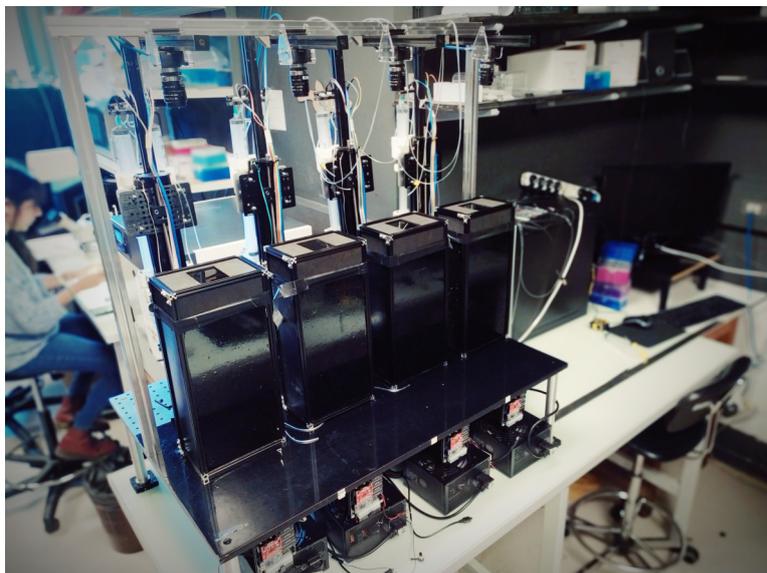


Figure 7.1 – Four Dual setups in parallel.

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

Mechanical

Dual's central mechanical part is a motorized syringe pump that creates the laminar flow. It is built around the V-Slot Linear Actuator from OpenBuilds fixed on a structure build using OpenBuilds linear rails and 3D printed fixations. A stepper motor with a gearbox (85:1) drives the actuator, and two microswitches limit the range of motion, see Figure 7.2.

It is crucial to isolate the animal from the exterior environment and of any light sources. A box that contains the millifluidic chip and lighting system is constructed using MakerBeam rails and medium-density fibreboard sheets. A PMMA infrared transparent sheet is placed on the removable top panel to record the experiment while blocking visible light. The box contains, see Figure 7.3, two LEDs for visible and infrared light, a diffuser for homogeneous lighting, and a support to fix the millifluidic chip.

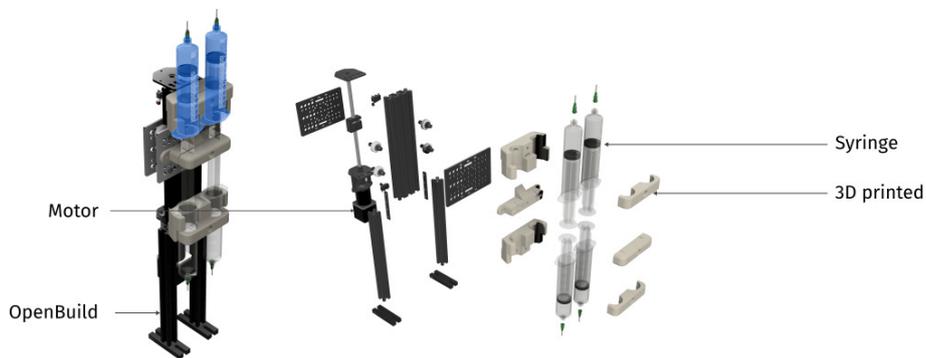


Figure 7.2 – Dual mechanical structure Dual custom pull-push syringe.

A structure to maintain the camera on top of the box and secure the Raspberry and the power alimentation is built using OpenBuilds rails. It is better to build Dual on top of an optical breadboard to facilitate fixation and enhanced stability. All these elements need to be fixed firmly and leveled to avoid any bias that could disturb the animal during the experiment.

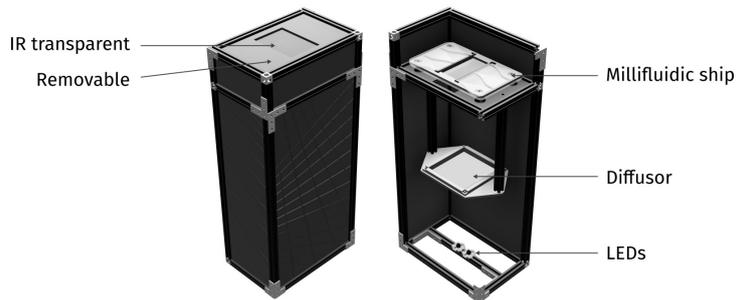


Figure 7.3 – Box The box isolates the animal from the exterior.

Millifluidic

The millifluidic chip that serves as a tank for the fish, see Figure 7.6, is laser cut in PMMA plastic that is transparent and presents excellent optic properties. The different parts are bonded using acetic acid [127]. The fish is restrained in the center by 3D printed and micro-machined nets, and profiled inputs and outputs allow a laminar flow.

Sixty-milliliters syringes are fixed with 3D printed fixations on the actuator and connected with 2.2 mm diameters tubing to the microvalves

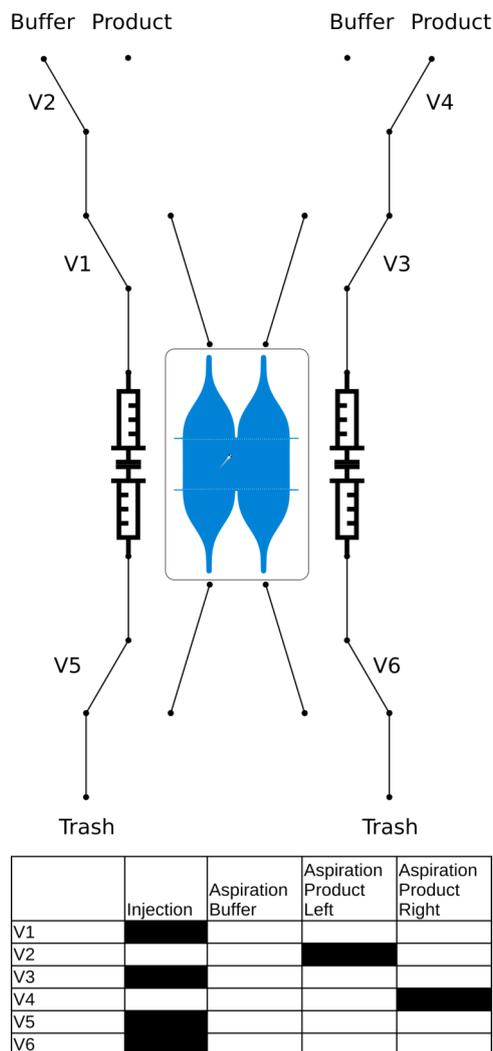


Figure 7.4 – Valves manifold Top: Valves system schematic. **Bottom:** Truth table of the system.

circuit. Six three-ports microvalves are connected and form a circuit that allows performing cycles of filling and injection, see Figure 7.4.

Imaging

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

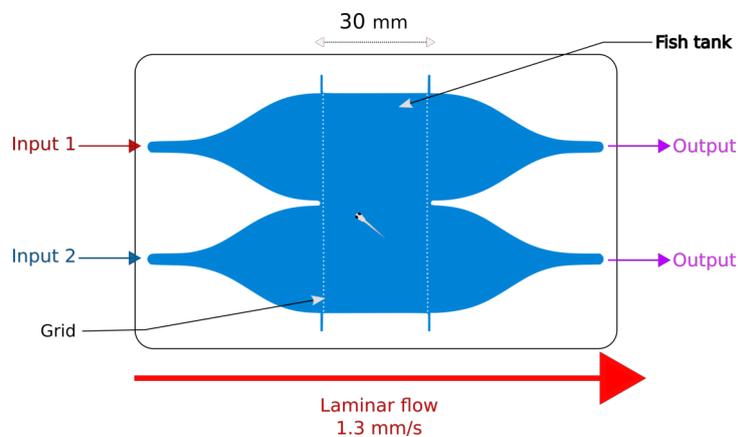


Figure 7.5 – Millifluidic chip schematic.

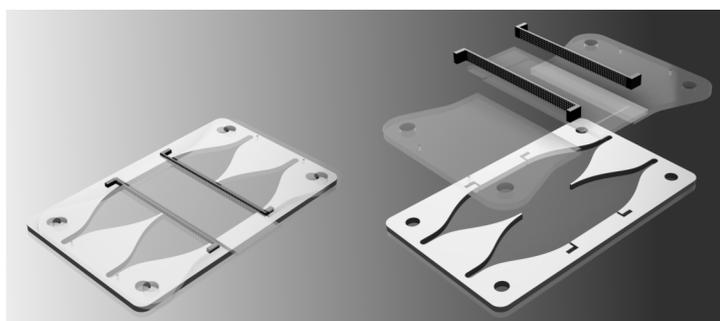


Figure 7.6 – Millifluidic chip exploded view.

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

Electronic

The electronic system links the analogic mechanical and millifluidic system to the software. A custom printed circuit board (PCB) has been designed for Dual. It contains an Arduino Nano microcontroller, six half H-bridges controlling the microvalves based on the logic signals of the Arduino; a Big EasyDriver stepper motor controller to control the motor speed from a logic output of the Arduino; a potentiometer to control the

intensity of the LED lighting. All the electronics and the syringe pump motor are powered by a 550W ATX computer power supply connected to the PCB.

Software

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

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

7.1.3 Construction and usage

For our needs, we have built four Duals that we ran in parallel, see Figure 7.1. The construction requires a laser cutting machine, a 3D printer, and workshop tools. It took about two weeks to build the four devices. The construction does not require any specific knowledge, and access to a FabLab is sufficient to carry out this project and find the required tools and help in the event of difficulties.

In practice, using Dual is effortless. Once the experiment template is created, the only manual task remaining is to place the fish in the millifluidic chip, close the device, and play the experiment template. It is also necessary to check that the suction containers neither run out of water or chemical. It is then possible to do experiments with minimal manual interventions.

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

In this project, we did not use all the Dual's potential. This setup can be adapted to other animals like adult *Danio rerio* a promising new model for neuroscience. It can assess other preferences, notably the preference to light (phototaxis) with the integrated visible LED.

7.2 The Tropical River

Studying chemical perception and chemically-driven navigation in a turbulent aquatic environment like the one fish encounters naturally requires an experimental device capable of creating controlled flows and chemical jets. We have created an experimental device capable of delivering a temperature-controlled laminar flow while recording the fish in both visible and infrared light. An injection nozzle is used to create turbulent or laminar jets within this laminar flow.

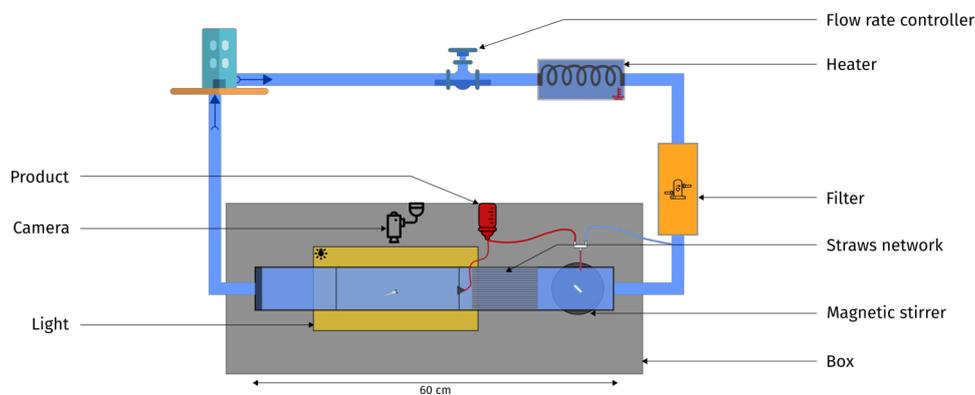


Figure 7.7 – The Tropical River Schematic of the experimental setup.

7.2.1 Description

Structure

The device's structure consists of a channel assembled from transparent polycarbonate sheets machined precisely by the mechanical workshop and fixed with Norcan and Makerbeam rails to form a channel (60x10x10 cm), see Figure 7.7. A LED panel is placed under the channel to illuminate the setup by transmission in visible light. The setup can be illuminated from above by a ring of infrared LEDs or by transmission by covering the LED panel with a PMMA infrared transparent sheet. A Chameleon 3 camera is placed above the channel to record the experiments and connected to the computer via USB3. A mirror is placed at 45 degrees of inclination from the horizontal on the channel's side to control the fish's vertical position. The whole channel is set inside a box constructed with Norcan rails and plywood sheets to isolate the fish from the surrounding environment.

Hydrodynamic

The canal is supplied at one end with water from the building's water system. Before entering the canal, tap water is filtered by an activated carbon filter and heated by a water bath. A network of straws is placed in the channel to obtain a laminar flow, and a solenoid valve can adjust

the flow rate. The other end of the channel is left free. The outgoing water is redirected to the building's wastewater network because the products tested do not require any special treatment before being disposed of. The water height inside the channel can be controlled by modulating the dyke height placed at the end of the channel.

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

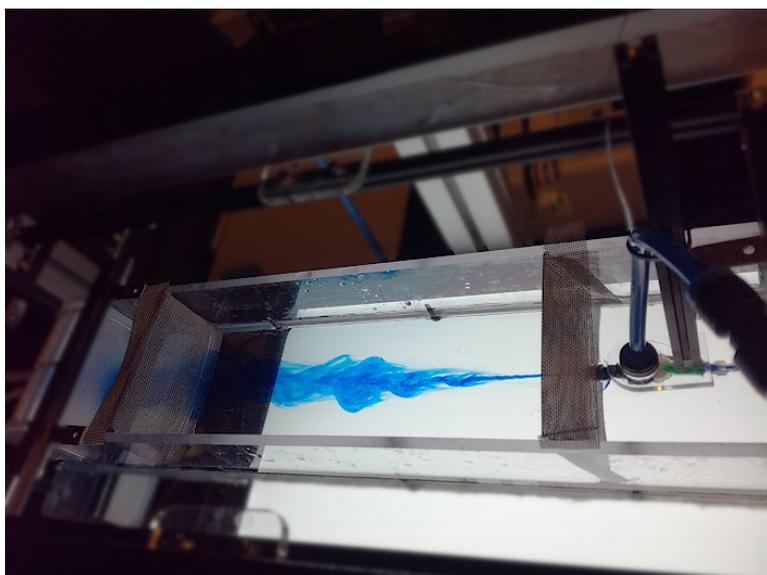


Figure 7.8 – The Tropical River Example of a turbulent jet visualized using blue visible colorant.

Software

The control software allows to retrieve and control all the variables of the experiment. By default, it allows selecting the flow rate, temperature, injection valves, and camera settings. A vital function of the software is the ability to build experimental protocols. Easy to build, the protocol

template is a simple text file specifying for a given device a variable's desired value at a given time point. Any Arduino sensor or control device that follows a convention detailed in https://github.com/LJPZebra/the_tropical_river_control can be called in these protocols without the need to modify the software.

Camera

The camera's options are directly accessible inside the software using the Spinnaker SDK. Metadata like temperature, relative times to the experiment, flow rate, or user-specified value can be saved inside the image.

Temperature

The temperature regulation is made using a coiled heat exchanger tube added to the water inlet and immersed into a Neslab RTE water bath capable of cooling and warming. A temperature sensor is placed inside the channel and sends the instantaneous temperature to the control software. The software can control the water bath via an RS232 serial connector selecting the temperature using a PID feedback system. Despite large variations of the building's water temperature, this system allows a precise and rapid temperature regulation.

7.2.2 Usage and limitation

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

The insertion of chemical in the flow does not allow us to reuse the

water. This is why the water supply is done using the water network of the building. Although filtered, the water quality depends on the building's water quality, which is usually not a problem for juvenile and adult fish that are reared in filtered tap water from 2 weeks onwards. However, this can be more problematic for larvae, which are more fragile and require calibrated water. Although the experiment's duration is limited only by the computer's storage, small air bubbles appear in the channel after a few hours. This phenomenon is due to the dissolved gases present in the tap water and is detrimental to the fish. The activated carbon filter reduces this phenomenon, but it remains present and limits the maximum experimental time to a few hours.

Chapter 8

Results

8.1 Methods

In the next sections, we will present the experimental protocol that we used and the analysis methods we developed to assess zebrafish chemical preference.

8.1.1 Experiment

Protocol Fish chemical preference was assessed using Dual setup in a one-hour long experimental in a protocol detailed in Figure 8.1. This protocol is subdivided into four cycles of 15 minutes each:

- B1: a cycle where buffer is injected on the two sides serving as a control.
- P1: a cycle where a product is injected on one side and buffer on the other.
- B2: same cycle as B1 in order to flush the system of any residual product.
- P2: same as P1 but with sides inverted.

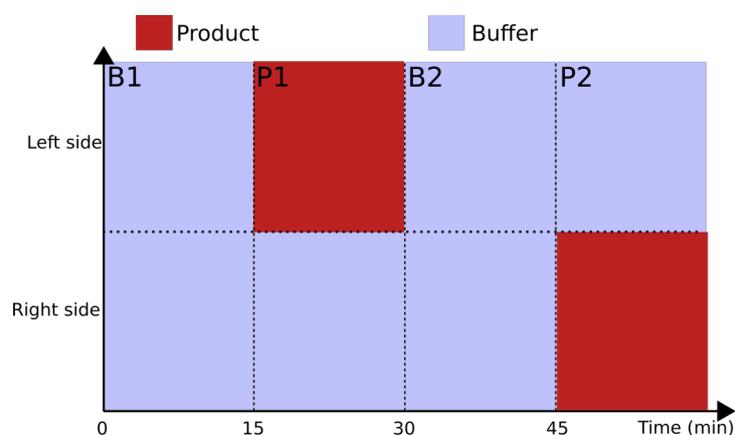


Figure 8.1 – Protocol Experimental protocol used to assess zebrafish chemical preference. Product cycles P1 and P2 were regularly inverted to avoid any side bias.

Fish In the following experiments, we use larval (6-8 days post-fertilization) and early juveniles (14 to 21 days post-fertilization) zebrafish. Preferences were assessed without changes in rearing condition before the experiment, except for ATP and adenosine, where fish were starved 24 hours before the experiment, a protocol inspired from [58]. Chemicals were dissolved in a buffer the day of the experiment: E3¹ for larvae and degassed tap water for juveniles.

Flow visualization All experiments were performed in the dark to isolate the fish from non-chemical perceptions, especially visual perception. The flow was visualized using an infrared dye: an emulsion of silicone oil, sodium dodecyl sulfate, and alginate prepared by Lea-Laetitia Pontani. This emulsion contains droplets with a diameter of $5\mu m$ with a polydispersity of $\approx 15\%$ and is biocompatible [128] at the concentration that we used ($\approx 500\mu L/L$).

¹<https://www.sc.niigata-u.ac.jp/biologyindex/natsuka/img/file26.pdf>

8.1.2 Analysis

The project's original idea was to use FastTrack (see Part 1) and an automatic custom-developed image analysis pipeline to monitor the fish preference using its position and recording the time spent in the product, see Appendix G.3.

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:

$$\Pi_{time} = \frac{t_{product} - t_{buffer}}{t_{product} + t_{buffer}} \quad (8.1)$$

When $\Pi_{time} = 1$, the fish spend all its time on the product side; thus, the product is attractive. When $\Pi_{time} = -1$, it is the complete opposite, and the product is repulsive. $\Pi_{time} = 0$ means that the fish spent the same amount of time on each side; thus, the product is neutral.

The critical point of this analysis method is to define when the fish is in the product. A coarse-grained analysis would be to assimilate the interface with its median position. Most of the time, it is accurate and robust to small inaccuracies. On the other hand, it will provide no insight at the crucial moment where the fish takes a decision, i.e., when the fish crosses the interface.

To precisely understand the fish's behavior, we needed to record these decision events with low fault tolerance. Counting the contact events automatically between the interface and the fish proved to be very challenging due to the fish's complex behavior, see Figure 8.2, and the complexity of extracting the interface position based solely on the dye's contrast. A large amount of data and time constraints pushed us to develop a manual analysis method to count these events. The time spent manually counting these events was shorter than developing an automatic algorithm, then checking the result.

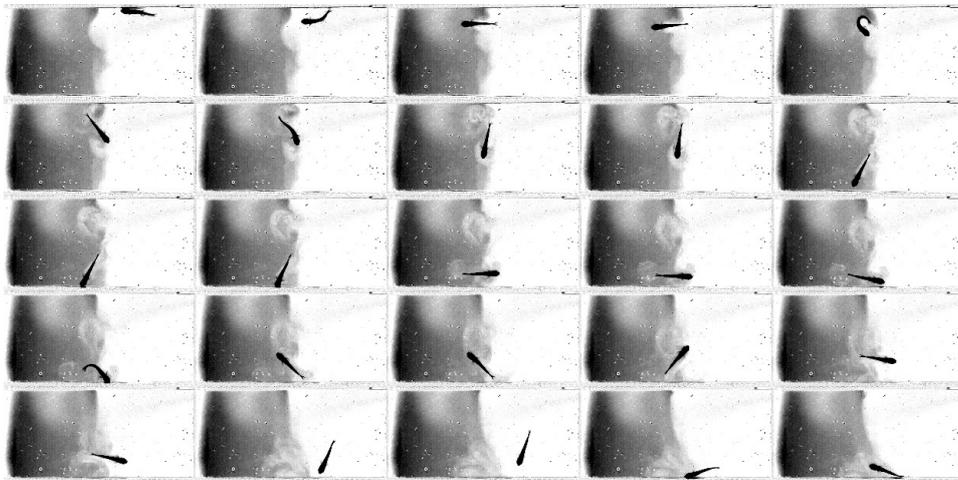


Figure 8.2 – Complex behavior Example of a complex behavior where the fish swim inside the interface. We see the product on the left side colored by the dye and the buffer on the right side. The fish is creating advection, and automatically extract the interface position is challenging. Pre-processed images where the background was subtracted to enhance the contrast between the product and the buffer.

Event analysis To develop the manual analysis method, we choose four characteristics and meaningful events to quantify the behavior, see Figure 8.3. When the fish cross the interface to change side (PB: product to buffer and BP: buffer to product) and when the fish sense the interface and return to the side where it was (BB: buffer to buffer, PP: product to product). By recording these four events, we can quantify the moment where the fish make a decision, which was not possible with the time-based analysis.

To count these events, we performed a blind manual analysis where movies were anonymized and scrambled. The events counting was performed by two independent individuals and deanonymized then pooled together to get the average.

We looked first at the correlation between the two independent analyses, see Figure 8.4, to check for human bias. The correlation between each fish's events count was equal to 0.96, indicating that there is no significant bias or disagreement between the analysis.

We can define an event-based preference index computed as the sum

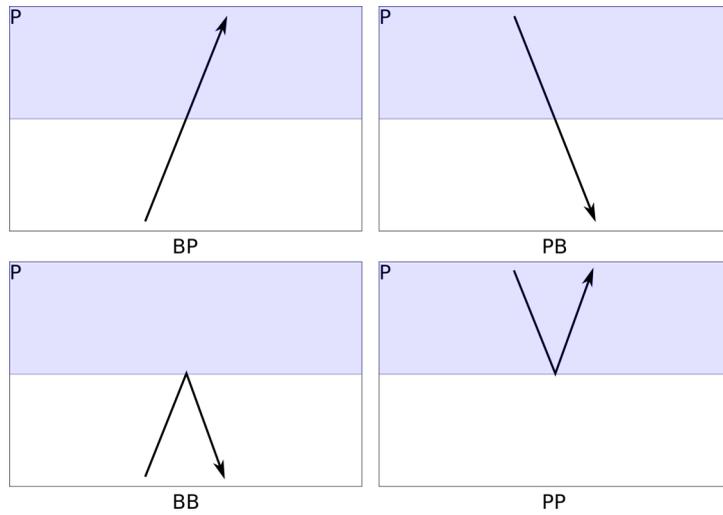


Figure 8.3 – Events The event-based method of quantification distinguishes 4 types of events that can occur during P1 and P2 product cycles.

of the events finishing in the product minus the sum of events finishing in the buffer, divided by the total number of events:

$$\Pi_{event} = \frac{n_{BP} + n_{PP} - n_{BB} - n_{PB}}{n_{BP} + n_{PP} + n_{BB} + n_{PB}} \quad (8.2)$$

This preference index is slightly different from the time-based one. It only reflects fish decisions and is not impacted by the fish inactivity and freezing.

Ratio exploration-exploitation Besides the preference of the fish, an interesting quantity that we can look at is the type of behavior. We can distinguish two modes of behavior: an exploration mode where the fish crosses the interface to explore the environment and an exploitation mode where the fish stays on the same side and makes touch-and-turn at the interface (see Appendix F for more details). Note that this quantity is not by any means a preference indicator.

The ratio exploration-exploitation can be computed based on the event recording as the number of events where the fish change side divided by the number of events where the fish stays on the same side:

$$\rho_{event} = \frac{n_{BP} + n_{PB}}{n_{BB} + n_{PP}} \quad (8.3)$$

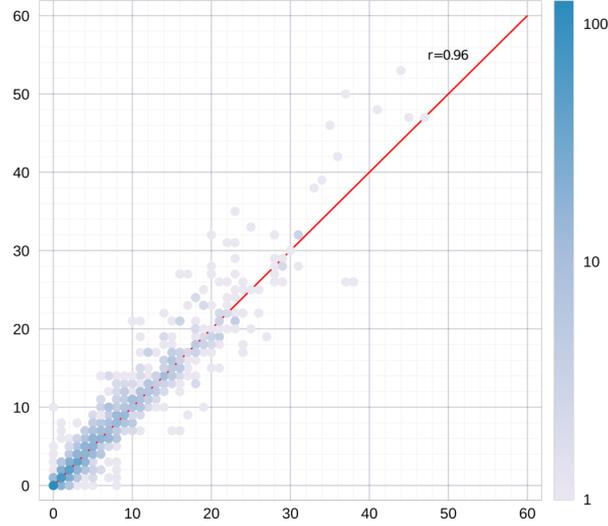


Figure 8.4 – Analysis correlation Correlation between the two independent event-based analysis. One point is the count for one event type for one fish. Red line is the identity line. The point color intensity is encoding the number of points at the same coordinate.

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

Markov model Based on the event recordings, we can build a two-state Markov chain model, see Figure 8.5. We can define the transfer probabilities as follows:

$$p = \frac{n_{BP}}{n_{BP} + n_{BB}}; b = \frac{n_{PB}}{n_{PB} + n_{PP}} \quad (8.4)$$

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

This model offers yet another definition of the preference index defined directly by

$$\Pi_{Markov} = p - b \quad (8.5)$$

the proportion of states product minus the proportion of states buffer. If p (resp. b) cannot be defined, we take $\Pi_{Markov} = 1 - 2b$ (resp. $\Pi_{Markov} =$

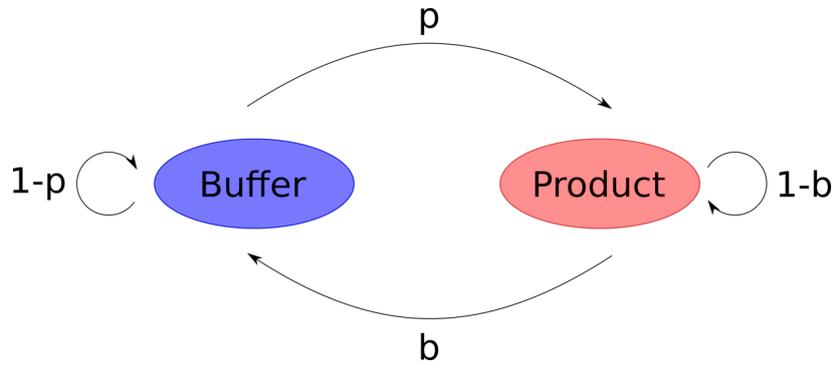


Figure 8.5 – Two states Markov chain model P indicates the product state and B the buffer state.

$-1 + 2p)$.

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

$$\rho_{Markov} = 2Min(p, b) - 1 \quad (8.6)$$

When $\rho_{Markov} = 1$, the behavior is purely exploratory, when $\rho_{Markov} = 0$ there is as many exploration than exploration and when $\rho_{Markov} = -1$ the behavior is dominated by exploitation.

We can build a numerical simulation of a two states Markov chain to explore the relationship between p and b probabilities, Π_{Markov} , and ρ_{Markov} . The numerical simulation produces sequences generated from a Markov chain defined with 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:

$$f(k) = \binom{k+n-1}{n-1} a^n (1-a)^k \quad (8.7)$$

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

We see in Figure 8.7 that there are forbidden couples of (p, b) due to the definition of the probabilities p and b as rational numbers and the limited chain lengths. The preference indexes are distributed on either side of the identity line $p = b$ where $\Pi_{Markov} = 0$ with an upper triangle repulsive and a lower triangle attractive. Looking at the map of ρ_{Markov} , we see a right upper corner at high p and b dominated by exploration, and a bottom left

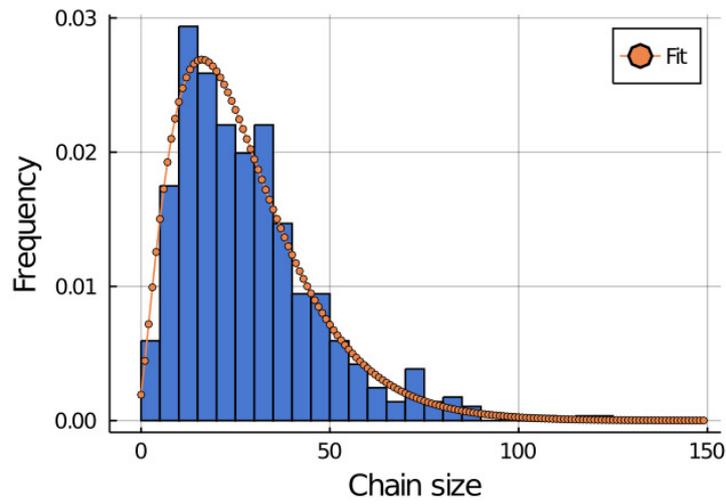


Figure 8.6 – Chain length Experimental chain length distribution.

corner by exploitation. As expected, the fish can have a neutral preference with either exploration or exploitation, but a strong preference can only happen in a regime dominated by exploitation.

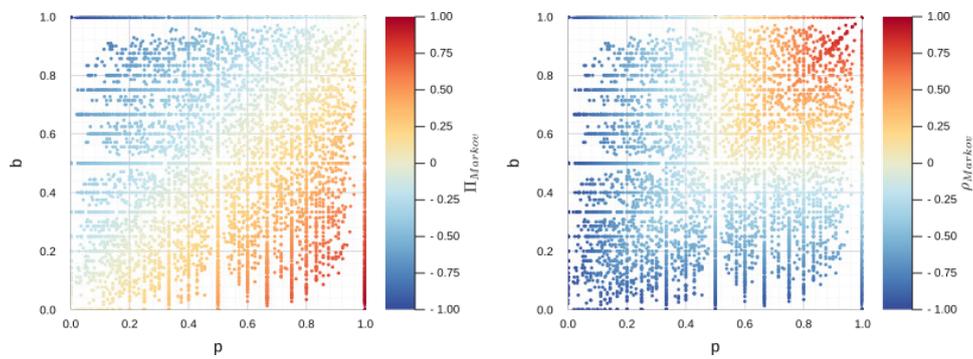


Figure 8.7 – Numerical simulation Preference index Π_{Markov} (left) and ρ_{Markov} (right) for 10 000 realisations with a chain size following the experimental distribution.

8.2 Results

8.2.1 Setup characterisation

Left-right bias We started by checking the preference index Π_{time} in the B1 buffer cycle to capture an eventual left-right bias of setup. There is buffer on the two sides in this control cycle, and fish were never exposed to any product. We computed the time-based preference index choosing the middle of the tank as left-right separation (in B1, there is no dye, thus no other means to define separation). Figure 8.8 presents the distribution of time-based preference index for B1 cycles for larvae and juveniles. The first thing we can notice is the significant variability among fish. The median value ($\Pi_{time} = 0.08$ for $N = 125$ for larvae and $\Pi_{time} = 0.05$ for $N = 178$ juveniles) is close to zero which allows us to exclude any systematic left-right bias of the setup.

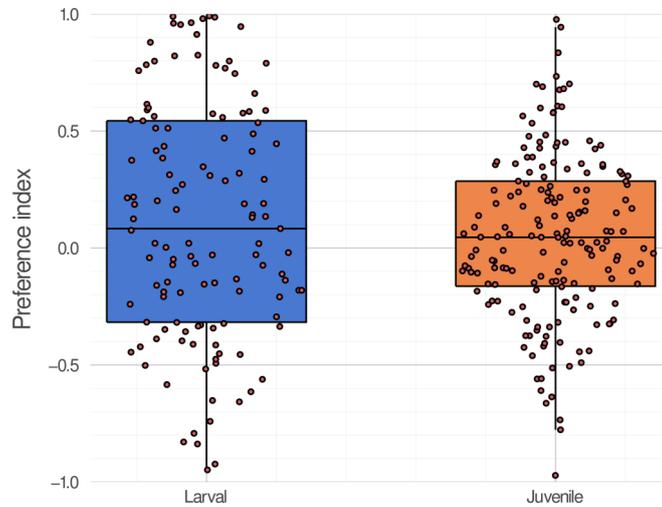


Figure 8.8 – Left-right bias Distribution of the time-based preference index Π_{time} computed for the first buffer cycle (B1) with buffer on the two sides. Each point is a value of preference index for one fish. Black line is the distribution median.

Dye neutrality In the product cycles P1 and P2, the dye is diluted with the product to visualize the flow. To study the dye's impact on the fish, we

performed control experiments with juvenile fish assessing only the dye with the protocol presented above.

Figure 8.9A presents the event-based preference index Π_{event} for the dye only. Each point is a fish analyzed by one person, and the point's size encodes the number of events during the cycle. We see that the dye seems neutral to the fish with no clear attraction or repulsion. As expected, there is large inter-fish variability. The preference index distributions for P1 and P2 are not statistically significant, but cycle P2 includes more outliers fish with strong preferences.

Figure 8.9B presents the time-based preference index Π_{time} , each point is representing one fish. We see a slight repulsion in the P1 cycle and a slight attraction in the P2 cycle towards the dye, but there is no clear, robust preference. These slight preferences are not visible with the event-based analysis. They are probably due to the time-based analysis's low precision with low statistics ($N = 13$) because it is more sensible to periods where the fish do not make any decision.

Figure 8.9C presents the p and b probabilities from the Markov chain model by fish. In each cycle, we see that $p \approx b \approx 0.65$ differs from 1, meaning that the fish can presumably perceive the dye or at least the side change by another unknown mean like a variation in flow velocity at the interface. This effect is also visible on the ratio exploration-exploitation ρ_{event} , see Figure 8.9D that is close to 1.

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

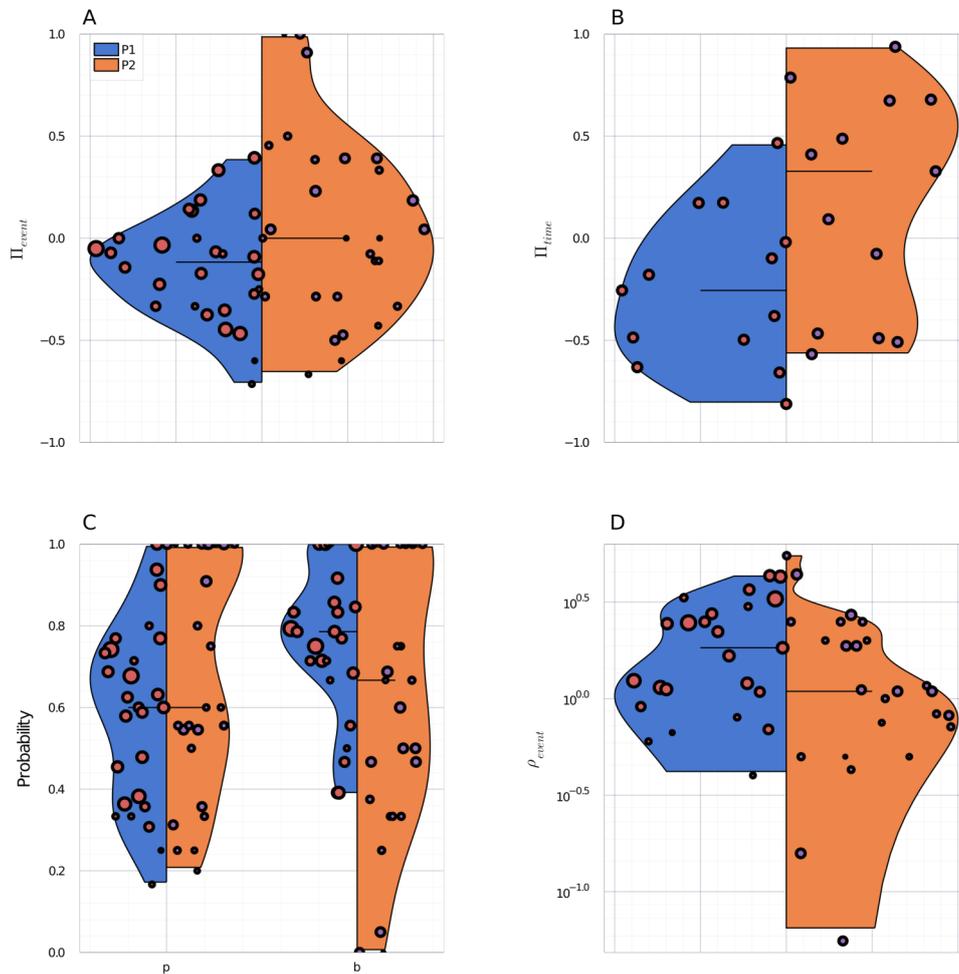


Figure 8.9 – Dye bias **A.** Event-based preference index Π_{event} , one point is one fish analysed by one person ($2N$ fish). **B.** Time-based preference index Π_{time} , one point is one fish. **C.** p and b probabilities from the Markov chain model, one point is one fish analysed by one person ($2N$ fish). **D.** Ratio exploration-exploitation ρ_{event} , one point is one fish analysed by one person ($2N$ fish).

8.2.2 Products screening

We have screened several products reputed to elicit an aversion or attraction with zebrafish. Our study focuses mainly on early juvenile (14 days) zebrafish with preliminary results for larvae (7 days).

Juvenile zebrafish in citric acid Citric acid is known to be repulsive in many fish species. It was shown that adult zebrafish could sense their environment pH [125,126] and display a strong aversion for acidic pH ($pH \approx 3$). As a positive control, early juveniles and larval zebrafish preference were assessed using citric acid solutions ranging from $5 \times 10^{-3} M$ to $1 \times 10^{-6} M$, ie pH from 2.8 to 4.2 (Figure 8.10).

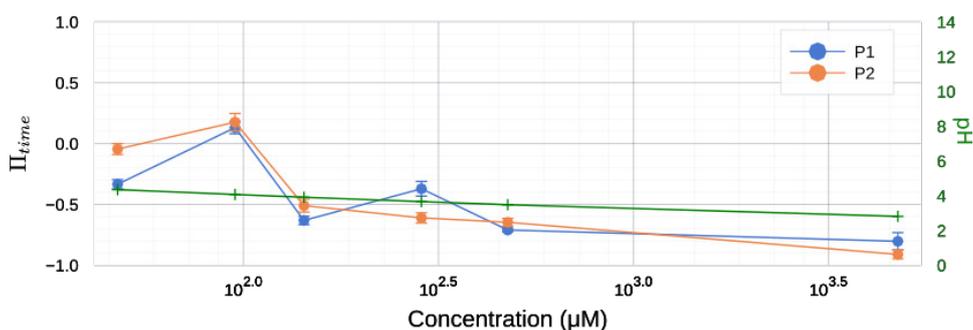


Figure 8.10 – Citric acid: time-based preference index for juveniles Time-based preference index (mean \pm SEM, equation G.1), calculated pH in green.

The mean ratios exploration-exploitation presented in Figures 8.11 bottom and Figure I.1 bottom, showed an apparent decrease when the citric acid concentration increase. Fish can sense citric acid and reduce their exploration to the benefit of a stereotyped exploitation behavior where they often go to the interface and return in the buffer.

The time-based Π_{time} (Figure 8.10 top), event-based Π_{event} (Figure 8.11 top), and Markov-based Π_{Markov} (Figure I.1 top) mean preference indexes decrease in a concentration-dependent manner. The distributions of preference indexes by fish (Figure J.1) show a decrease in variability as the concentration increase and no significant difference between P1 and P2 cycles, which tends to suggest that fish adopt the same characteristic behavior with less variability as a preference emerged.

We tried to see if the bulk-behavior (i.e., modulation of the kinematic parameters inside the product) was able to explain the time-based preference index. We extract the parts of trajectory inside the product and the buffer during the P1 and P2 cycle and inside buffer in the B1 cycle as a control. Larvae and juveniles swim in units called bouts separated by rest periods called interbouts. We discretized the trajectory by swim-bouts and

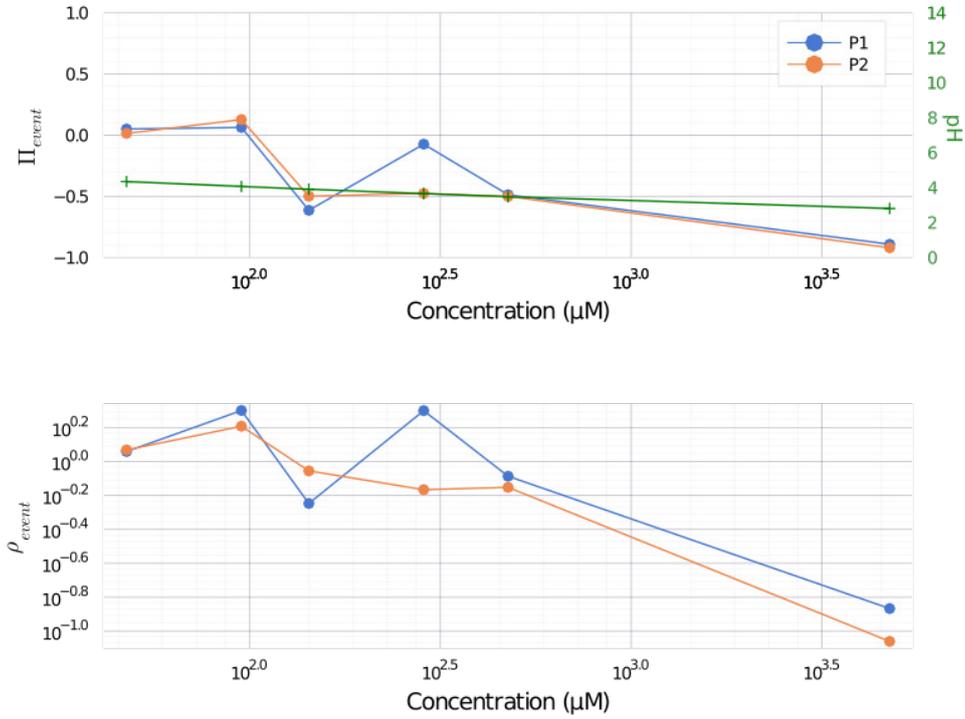


Figure 8.11 – Citric acid: event-based analysis for juveniles Top: Mean event-based preference index (equation G.2). **Bottom:** Mean ratio exploration-exploitation (equation G.3).

extracted three kinematic parameters: the time δt , the distance δr , and the angle $\delta \theta$, between the onset of successive bouts.

It was shown by [129] that the mean escape time of a domain with a mix of reflective and absorbing walls by a random walker moving with constant velocity and changing directions at random times writes :

$$\langle t \rangle = \alpha \frac{V}{vS} \quad (8.8)$$

with V the volume and S the surface of the domain, α a numerical coefficient, and v the velocity of the random walker.

In our case, Dual can be seen as a domain of characteristic size a with reflective walls (Dual edge) and an absorbing wall (the interface between product and buffer), see Figure 8.12 left. The fish trajectory can be seen as a random walk with a reorientation at each swim-bout. We see on Figure 8.12 right that δt and δr are uncorrelated and that there is a timescale

τ and a lengthscale λ such as:

$$\langle v \rangle = \frac{\lambda}{\tau} \quad (8.9)$$

and with Equation 8.8, 8.9 and $\frac{V}{S} = a$ in Dual we have:

$$\langle t \rangle \approx \alpha \frac{a\tau}{\lambda} \quad (8.10)$$

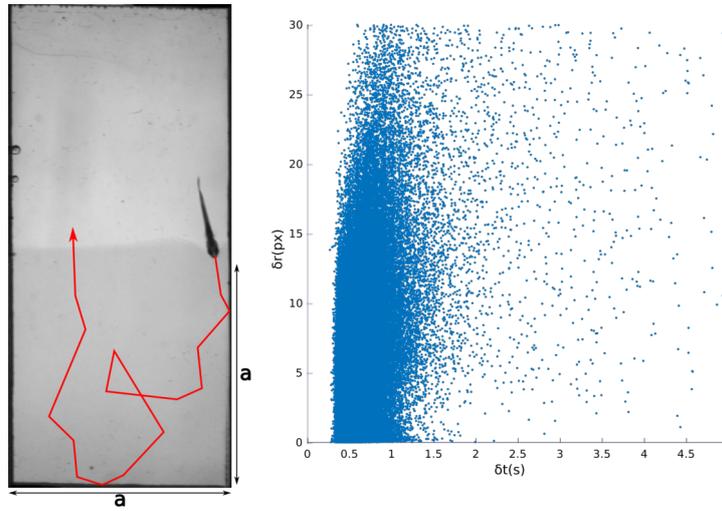


Figure 8.12 – Bulk-behavior characterization Left Fish trajectory inside Dual, one side is a square of size a with 3 reflective walls and one absorbing wall the interface. **Right:** Correlation between δt and δr .

The preference index can be computed as the mean time spent in the product minus the mean time spent in the buffer divided by the total time. Therefore, from Equation 8.10 we expect no influence of the distribution of $\delta\theta$ in the generation of a preference index.

In Figure H.1, we see no difference between the distributions when we increase the concentration in citric acid. We built a numerical simulation by generating random trajectories using the experimental distributions and computed the time-based preference index, see Figure 8.13. We see that the bulk-behavior cannot explain the preference index and that decision events at the interface are crucial in generating preference.

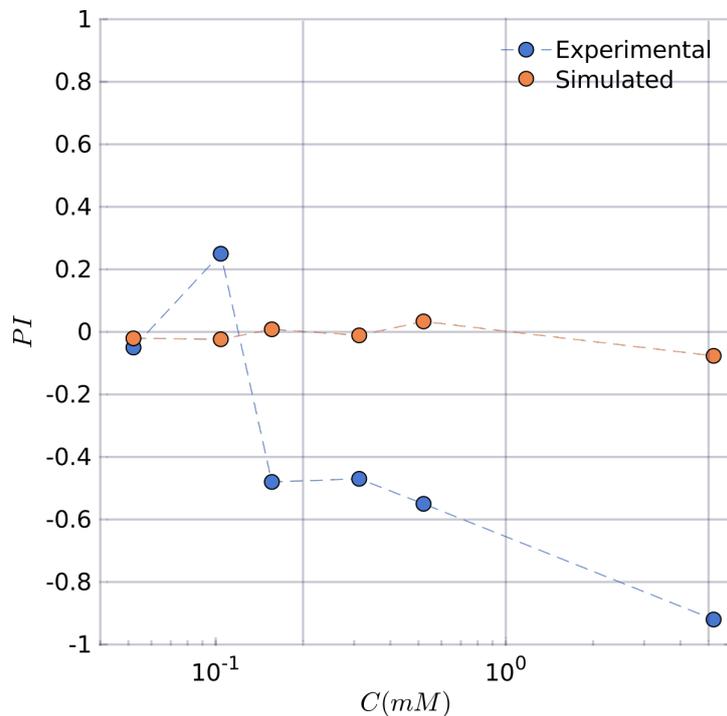


Figure 8.13 – Preference index numerical simulation

Zebrafish larvae in citric acid Preliminary experiments were done to assess the effect of citric acid on zebrafish larvae (Figure 8.14). Despite low statistics (see Section 9) and substantial variability, we see that citric acid seems to be repulsive as expected from results with juveniles. More experiments need to be done to confirm this effect and account for the larvae variability.

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

Juvenile zebrafish in quinine Quinine is a natural alkaloid that tastes bitter, and that is a highly deterrent substance for many fish species [130]. Unexpectedly, we see with the time-based preference index (Figure 8.15) that the preference seems to be mostly neutral. A slight attraction can be

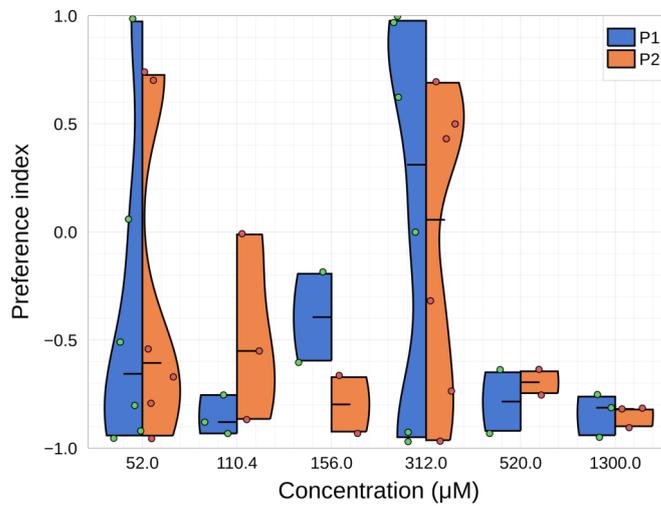


Figure 8.14 – Citric acid: preference index for larvae Time-based preference index, one point is one fish.

constated on the P1 cycle, and a neutral a slightly repulsive preference on the P2 cycle.

Despite relatively good statistics (≈ 7 fish by concentration), no robust preference emerged as we see with large variability in responses.

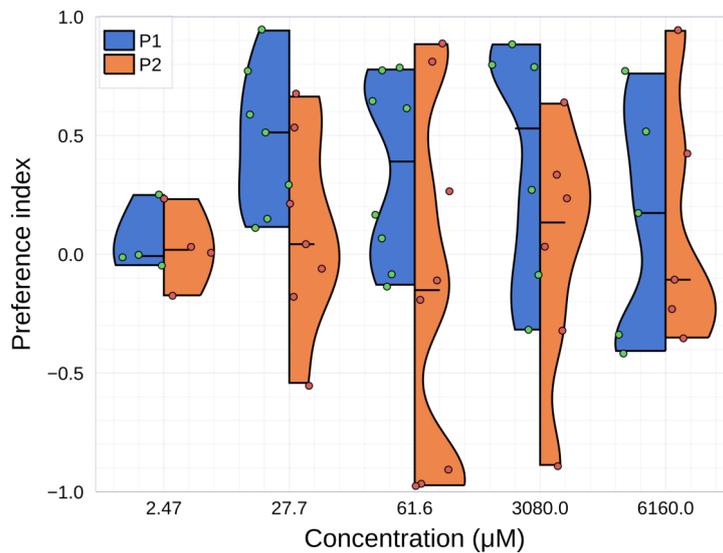


Figure 8.15 – Quinine: preference index for juveniles Time-based preference index. One point is one fish.

ATP and adenosine Zebrafish have pear-shaped olfactory sensory neurons expressing olfactory receptors specific to adenosine [58]. ATP, ADP, and AMP are dephosphorylated in the zebrafish olfactory epithelium [58] and behavioral responses to adenosine and ATP were shown to exist in adults with arousal and food-seeking behavior. We assessed preference for ATP and adenosine for larval and early juvenile zebrafish. Fish were starved 24 hours before the experiment, and preference was assessed using the protocol presented above.

Juvenile zebrafish in adenosine Fish can perceive adenosine as indicated by the ratios exploration-exploitation presented in Figures 8.17 bottom and I.2 bottom that decrease with the concentration.

The event-based and Markov-based preference indexes (Figures 8.17 top and I.2 top) show a slight repulsion for the P1 cycle and a slight attraction for the P2 cycle at the two lower concentrations. The time-based preference index (Figures 8.16) displays a slight repulsive preference for the P1 cycle. There is no clear preference, and analysis from event-based and time-based preferences disagree, meaning that there is probably not enough statistic to conclude. Besides, the preference index distributions (Figure J.2) show great variability.

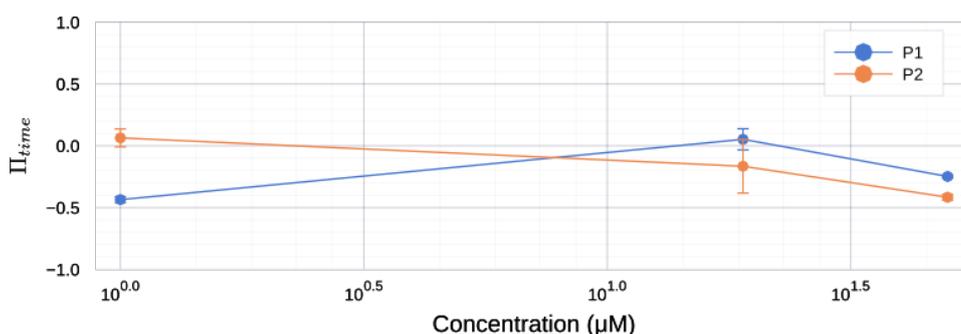


Figure 8.16 – Adenosine: time-based preference index for juveniles Time-based preference index (mean \pm SEM, equation G.1).

As no clear preference emerged from the early experiments, we focused on the concentration $50\mu M$ and increased the statistics up to $N = 28$. We then saw a slight repulsion on the P1 cycle at the same level as the other concentrations. However, the P2 cycle showed a clear repulsion

with a median preference index equal to -0.6 and 2 clusters of fish, one attracted and one repulsed.

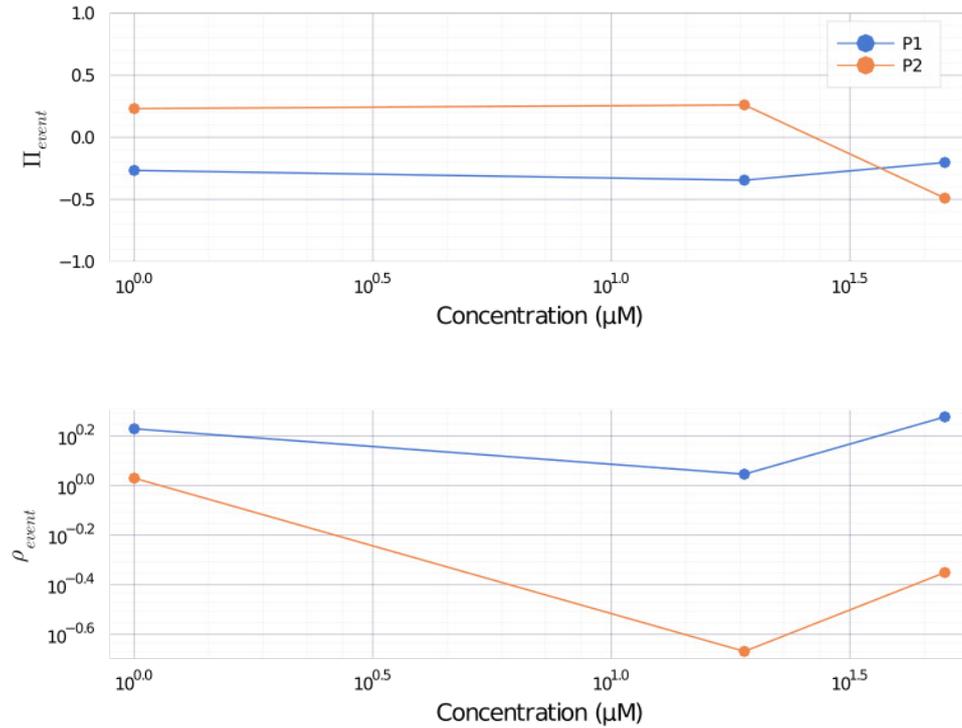


Figure 8.17 – Adenosine: event-based analysis for juveniles **Top:** Mean event-based preference index (equation G.2). **Bottom:** Mean ratio exploration-exploitation (equation G.3).

Unexpectedly, adenosine does not seem to attract early juvenile zebrafish. On the contrary, at high concentration, it is clearly repulsive. However, one part of the fish ($\approx 30\%$) seems to be either attracted or neutral at the second presentation of the product.

Zebrafish larvae in adenosine Zebrafish larvae were exposed to adenosine with the same protocol as juveniles. Preliminary results presented in Figure 8.18 and based on the time-based preference index seem to indicate a lack of preference at all concentrations.

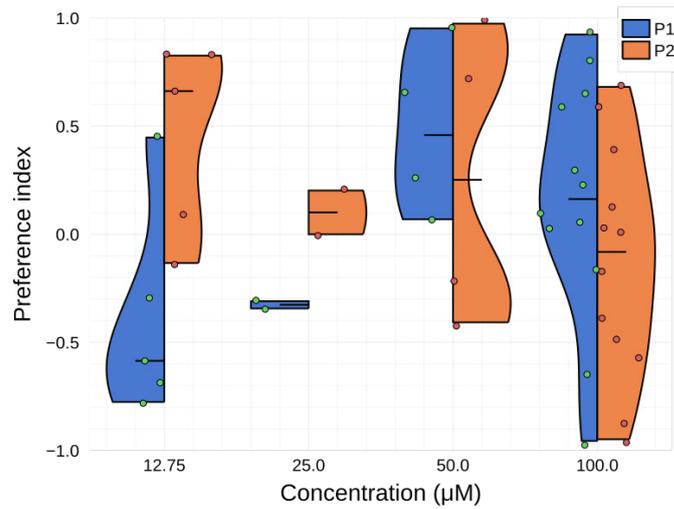


Figure 8.18 – Adenosine: preference index for larvae Time-based preference index. One point is one fish, black line is the distribution median.

Juvenile zebrafish in ATP Interestingly, we found that ATP, a product that was shown to be attractive on adults, produces a two phases behavior on early juvenile zebrafish (Figures 8.19 top, 8.20 top, and I.3 top). At low concentration, the fish do not manifest a preference, but at the highest concentration ($125\mu M$ and $200\mu M$), the P1 cycle is repulsive and the P2 cycle attractive.

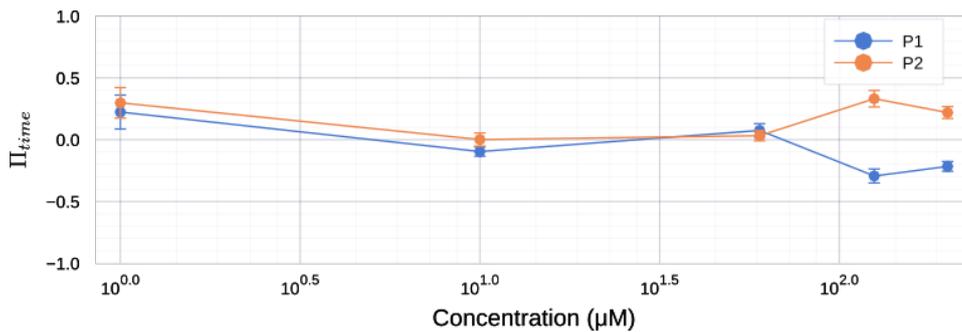


Figure 8.19 – ATP: time-based preference index for juveniles Time-based preference index (mean \pm SEM, equation G.1).

Comparing the distribution of event-based, Markov-based, and time-based preference index by fish, we see that this effect is significant at $125\mu M$ and $200\mu M$, see Figure J.3.

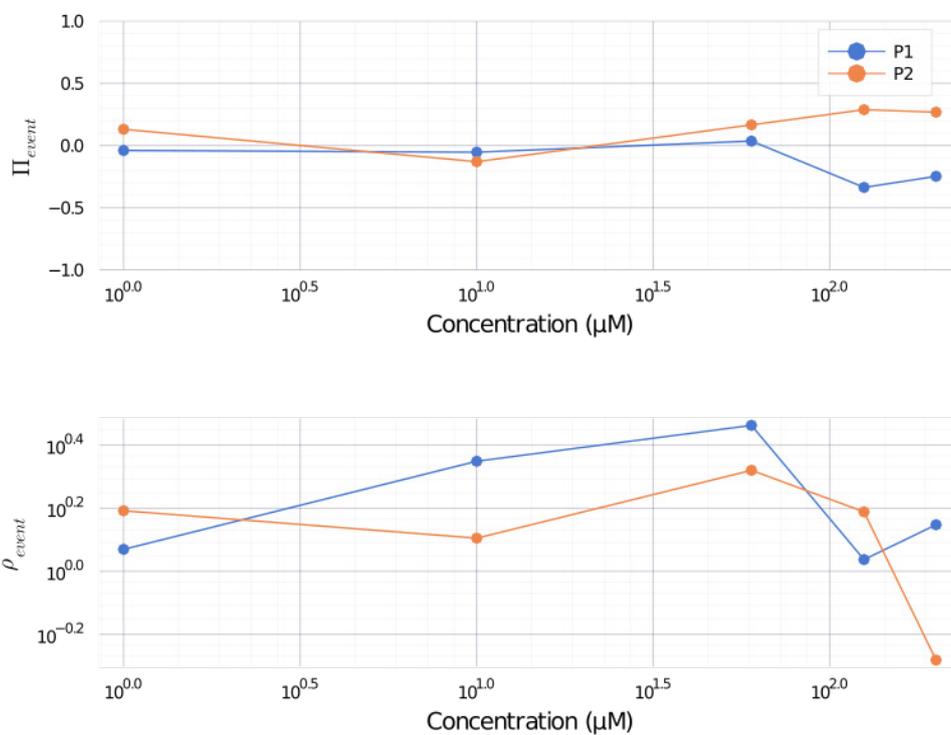


Figure 8.20 – ATP: event-based analysis for juveniles **Top:** Mean event-based preference index (equation G.2). **Bottom:** Mean ratio exploration-exploitation (equation G.3).

A fish-by-fish analysis, see Figure 8.21A, revealed that the fish proportion that changes its preference towards attraction increase with the concentration. Figure 8.21B represents the p and b probability from the Markov chain model in function of the cycle P1 and P2. We see that many fish cross the identity line between the P1 and P2 cycle to invert their preferences.

Zebrafish larvae in ATP Zebrafish larvae were exposed to ATP with the same protocol as juveniles. Preliminary results presented in Figure 8.22 and based on the time preference index indicate a repulsion at all concentrations. High concentrations seem to produce more robust repulsive behavior with low variability.

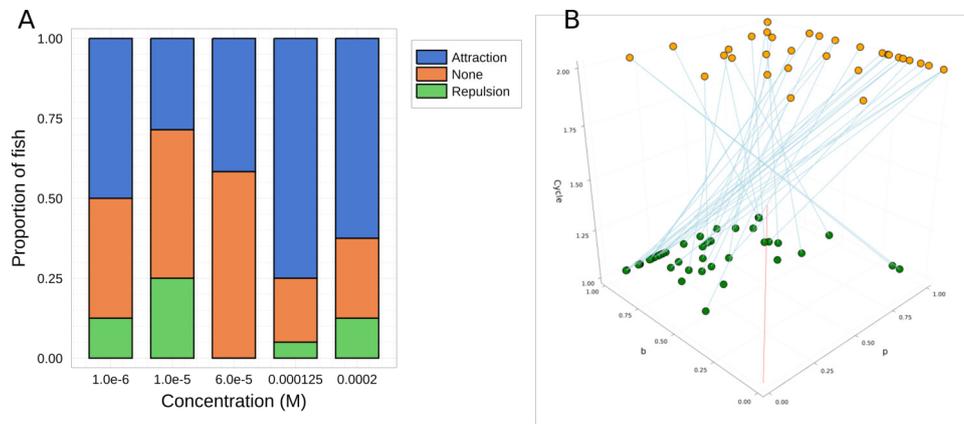


Figure 8.21 – Cycle effect A. Fish proportion changing preference between the P1 and P2 cycles. **B.** Probabilities p and b for 125 and 200 μM , one point is one fish analyzed by one person ($2N$ fish). Cycle are represented by the point's altitude.

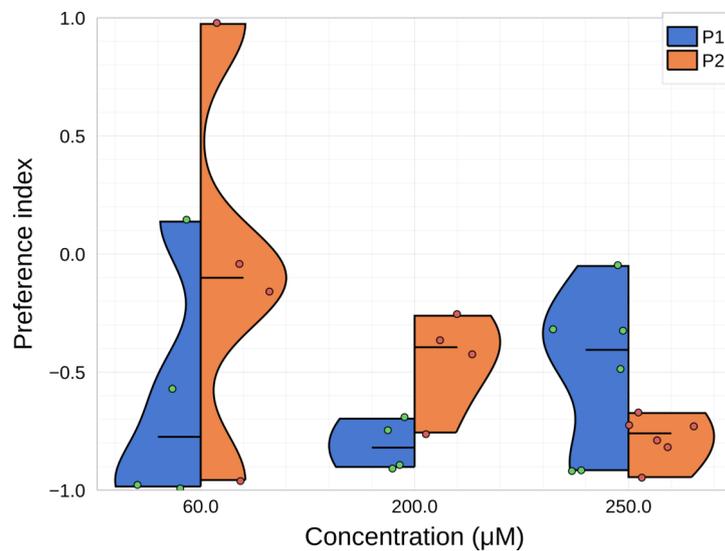


Figure 8.22 – ATP: preference index for larvae Time-based preference index.

Chapter 9

Discussion

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

The Tropical River is a setup capable of producing controlled flows. The flow temperature and velocity can be precisely controlled. Turbulent and laminar jets can be created inside the flow to mimic more realistic fragmented chemical perception. This setup was initially developed to study the chemically-driven navigation after finding a robust, attractive compound with Dual. Finding such a product was challenging because it requires many statistics, thus many experiments. We have concentrated our efforts on this task with Dual; therefore, we could not use The Tropical River in this project's context. Nonetheless, several preliminary experiments were performed to check the setup. It was used in another project by Bill François to study fish swimming taking advantage of the flow velocity control and high framerate imaging.

Dual is a high-throughput chemical screening setup with a high level of versatility that allows testing several combinations of products, concentrations, and fish ages easily and rapidly. The setup is scalable and can be built at the laboratory for less than 2 000 euros. In the first part of the thesis, we built the setup and controlled that it did not have any bias. We also checked that the infrared dye used to visualize the flow was neutral

to the fish.

Then we developed several analysis methods to capture the fish preference and behavior. These methods are fragmented according to fish ages and products. We first implemented a coarse-grained approach where we recorded the fish's position and computed the time-based preference index taking the median interface position. Then we tried to refine this approach to better characterize the fish behavior at the interface by extracting the interface position. This was not practically achievable due to several constraints: time, fish complex behavior, and challenging image processing problems on a large amount of data (8 To of data). Finally, we chose to perform a manual analysis that focuses on events happening at the interface, moments where the fish had to make a decision. It was possible to confirm the time-based analysis results with this approach and characterize the fish behavior more precisely. This method was only applied to products with enough statistics and already interesting effects to delve into.

We successfully used Dual to assess the chemical preference of several products. We show a strong aversion to citric acid and an exciting effect with ATP where fish inverts their preferences at the second product presentation.

Adults were shown to be attracted by ATP without mention of prior exposures. Our results tend to suggest that larvae are repulsed and juveniles repulsed then attracted in a second exposition to ATP. This behavior is probably more complicated because we see variability among fish, with fish changing preference and others not. Work remains to be done to precisely characterize this effect. It would be interesting to study the proportion of fish switching preference by, for example, forcing ATP on both sides for a given duration and then check preference as a function of the first presentation time or the first presentation concentration. An interesting question would be to check what happened at the third presentation of ATP. All these experiments can be done easily by modifying the protocol in Dual. If this attractive preference persists in time, ATP could be used in The Tropical River to study chemically-driven navigation. A temporal characterization would also be of great interest. Studying the effect at a fixed

concentration, like $125\mu M$, and varying the fish age. It would help understand the development of ATP's perception from larval to adult stage.

Preliminary results shown with larvae need more statistics. Studying larvae poses several challenges. First, larvae tend to explore less than juveniles, and combined with their small sizes, they do not frequently cross the interface. We circumvented this problem by putting up to 4 larvae by experiment, thus increasing the statistics, but this adds work in the analysis phase. Secondly, larvae are more susceptible to fatigue and freezing than juveniles. It frequently happens that larvae stop swimming and do not recover until the end of the experiment. For this reason, assessing the ATP effect on larvae is very challenging: reducing cycle length to reduce fatigue would lead to fewer crossing events, but longer cycles would lead to more fish dropouts before the second cycle. Another solution would be to reduce the chamber size to increase crossing events with the same cycle length. For all these reasons, studying larvae would need many statistics, which we had not the time to do in this thesis.

Studying the chemical perception of zebrafish proved to be challenging. The fish inter-variability is high, as we have seen by screening products. In these experiments, we use four Duals to record up to 350 hours of usable experiments, but we see that more statistics are needed. We hope that Dual as a low-cost, easily replicable setup will encourage people to build and use the setup to better characterize zebrafish chemo-preference.

Appendices

Appendix E

Dual bill of materials

Category	Manufacturer	Description	Link	Qty	U.P.
Computer	Raspberry Pi	RASPBERRY PI 4B/4GB	https://www.digikey.fr/product-detail/en/raspberry-pi	1	44.84 €
	Raspberry Pi	RPI USB-C POWER SUPPLY BLACK EU	https://www.digikey.fr/product-detail/en/raspberry-pi	1	6.52 €
	Raspberry Pi	RASPBERRY PI 7" TOUCH SCREEN LCD	https://www.digikey.fr/product-detail/en/raspberry-pi	1	48.92 €
	Corsair	Flash Voyager GTX USB 3.1 128 Go	https://www.amazon.com/gp/product/B079NVJPKV/	1	45.57 €
Optics	FLIR	Chameleon3 Camera (CM3-U3-13Y3M-CS)	https://www.flir.eu/products/chameleon3-usb3/?mode	1	285.00 €
	FLIR	USB 3.1 Locking Cable	https://www.flir.eu/products/usb-3.1-locking-cable/	1	8.00 €
	Thor Labs	MVL25M23 - 25 mm EFL, f/1.4, for 2/3" C-Mount	https://www.thorlabs.com/thorproduct.cfm?partnumber	1	192.80 €
	GoodFellow	ME303007 PMMA 0,5mm - IR transparent (80x60cm rectangle + Ø20mm disk)	http://www.goodfellow.com/catalogue/GFCat41.php?	1	12.62 €
	Mouser	LED infrarouge 897-LZ440R608	https://www.mouser.fr/ProductDetail/LED-Engin/LZ4	1	17.69 €
		Coolbox	ATX Power supply 500W Basic	https://www.amazon.fr/CoolBox-Basic-500GR-dalim	1
PCBWay		Custom PCB	https://www.pcbway.com/	1	3.93 €
Arduino		Arduino nano 3.1	https://www.mouser.fr/ProductDetail/DFRobot/DFRO	1	16.52 €
Sparkfun		Big easy driver	https://www.mouser.fr/ProductDetail/SparkFun/ROB	1	16.90 €
Sparkfun		ATX Power Supply Connector - Right Angle	https://www.mouser.fr/ProductDetail/SparkFun/PRT-	1	1.65 €
New Japan Radio		NJM2670D2	https://www.mouser.fr/ProductDetail/NJR/NJM2670	6	2.76 €
Lite-on		LED through-hole green	https://www.mouser.fr/ProductDetail/Lite-On/LTL2R	7	0.09 €
Lite-on		LED through-hole yellow	https://www.mouser.fr/ProductDetail/Lite-On/LTL2R	4	0.09 €
Bourns		Potentiometer 1 kohms	https://www.mouser.fr/ProductDetail/Bourns/PTV09A	2	0.68 €
Phoenix contact		2-wires connector	https://www.mouser.fr/ProductDetail/Phoenix-Contac	8	1.29 €

Appendix F

Touch-and-turn behavior

Strong preferences are always found in a regime where exploitation dominate. It can be easily observed by a behavior that we call touch-and-turn, see Figure F.1. The fish goes to the interface to sense it, then abruptly turns to return to its preferred side.

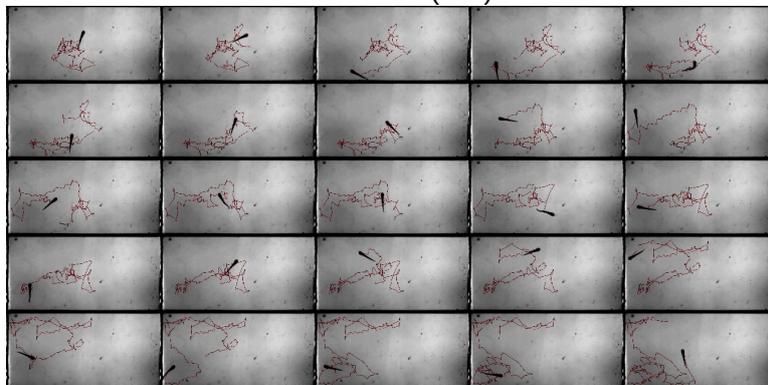
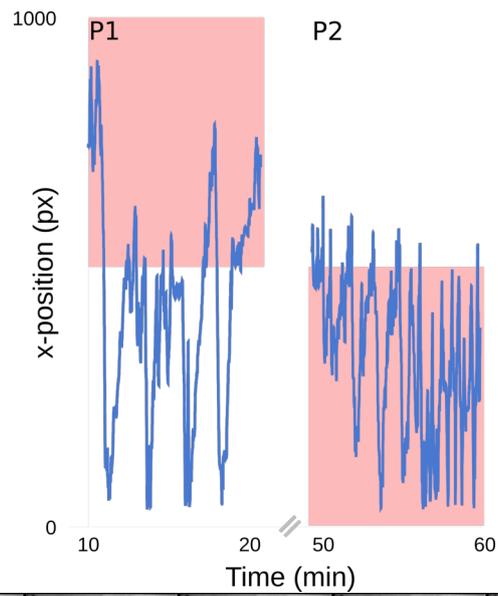


Figure F.1 – Touch-and-turn behavior Left: Fish trajectory in $200\mu M$ of ATP (red box).

Appendix G

Calculation

G.1 Mean quantities

Mean quantities were calculated from events count following these definition:

$$\overline{\Pi_{time}} = \frac{\sum_{fish}(\Pi_{time})}{N_{fish}} \quad (G.1)$$

$$\overline{\Pi_{event}} = \frac{\sum_{fish}(n_{BP} + n_{PP} - n_{BB} - n_{PB})}{\sum_{fish}(n_{BP} + n_{PP} + n_{BB} + n_{PB})} \quad (G.2)$$

$$\overline{\rho_{event}} = \frac{\sum_{fish}(n_{BP} + n_{PB})}{\sum_{fish}(n_{BB} + n_{PP})} \quad (G.3)$$

$$\bar{p} = \frac{\sum_{fish} n_{BP}}{\sum_{fish}(n_{BP} + n_{BB})} \quad (G.4)$$

$$\bar{b} = \frac{\sum_{fish} n_{PB}}{\sum_{fish}(n_{PB} + n_{PP})} \quad (G.5)$$

$$\overline{\Pi_{Markov}} = \bar{p} - \bar{b} \quad (G.6)$$

$$\overline{\rho_{Markov}} = 2Min(\bar{p}, \bar{b}) - 1 \quad (G.7)$$

G.2 Statistical tests

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

G.3 Image analysis

Fish head position was extracted using FastTrack. Each recording was manually checked to remove low-quality movies either with imperfect flows or non-responsive fish. Swim-bouts were extracted using the velocity signal and a threshold.

A custom image processing procedure was developed to extract the interface using the contrast difference due to the dye ¹. First, the maximal and minimal z-projection of the movie was calculated. For the minimal projection, the fish is masked with a white mask on each image before the projection, only keeping the flow and not projection the fish. Each image of the movie is then normalized with minimal and maximal projection. Finally, the interface was detected using an Otsu threshold.

A python command-line interface ² was created to export the relevant data in a synthetic toml file easy exportable, one experiment by file. Statistical analysis and numerical simulations were performed using Python and Julia programming language.

¹https://github.com/LJPZebra/dual_analysis/tree/master/dual_cli

²https://github.com/LJPZebra/dual_analysis

Appendix H

Kinematic parameter distributions

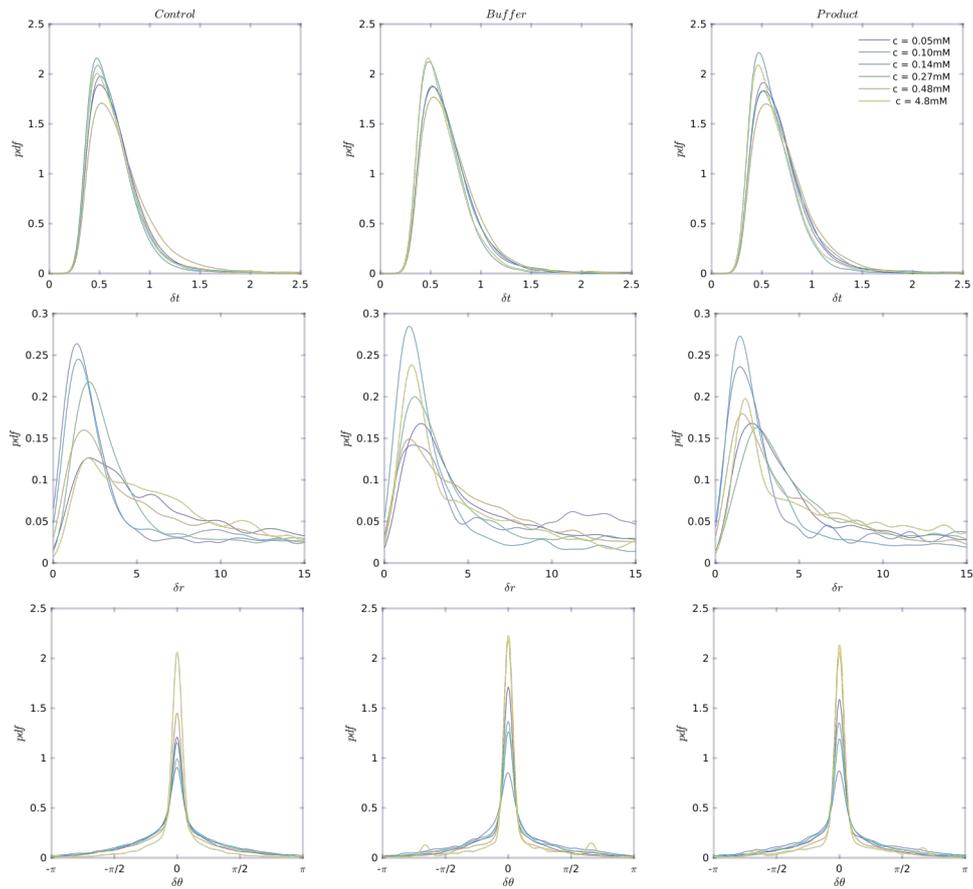


Figure H.1 – Distributions of kinematic parameters for juveniles in citric acid. Top: δt time between on the onset of successive bouts. **Middle:** δr distance between on the onset of successive bouts. **Bottom:** $\delta \theta$ angle between on the onset of successive bouts. Control means kinematic parameters extracted from the B1 cycle, buffer (resp. product) are extracted from P1 and P2 cycles when the fish is in the buffer (resp. product).

:

Appendix I

Markov model results

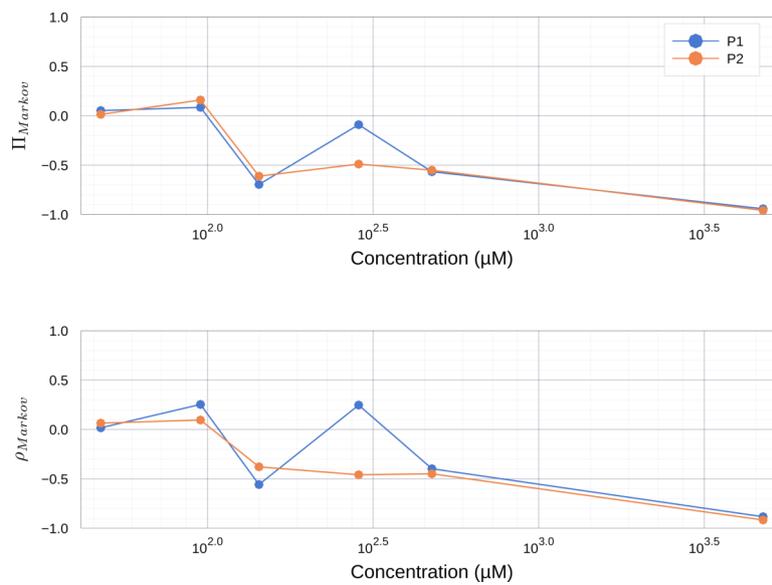


Figure I.1 – Citric acid: Markov chain analysis for juveniles **Top:** Mean Markov-based preference index (equation G.6). **Bottom:** Mean Markov ratio exploration-exploitation (equation G.7).

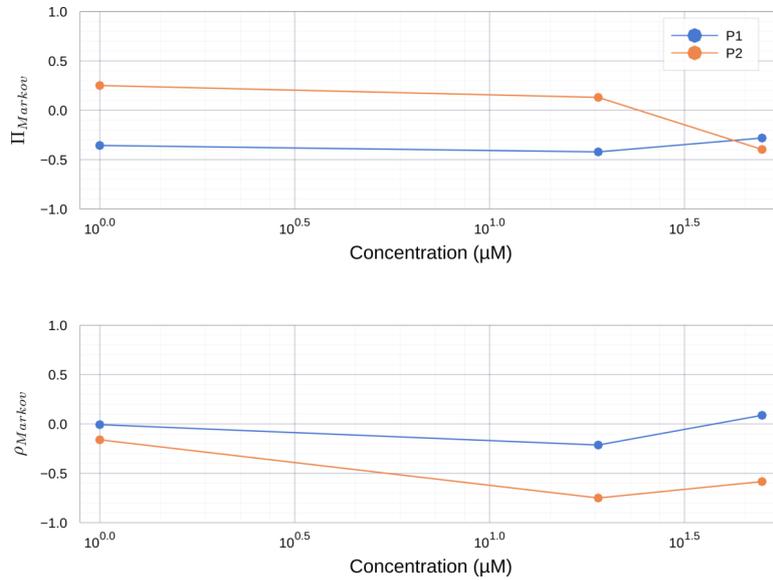


Figure I.2 – Adenosine: Markov chain analysis for juveniles **Top:** Mean Markov-based preference index (equation G.6). **Bottom:** Mean Markov ratio exploration-exploitation (equation G.7).

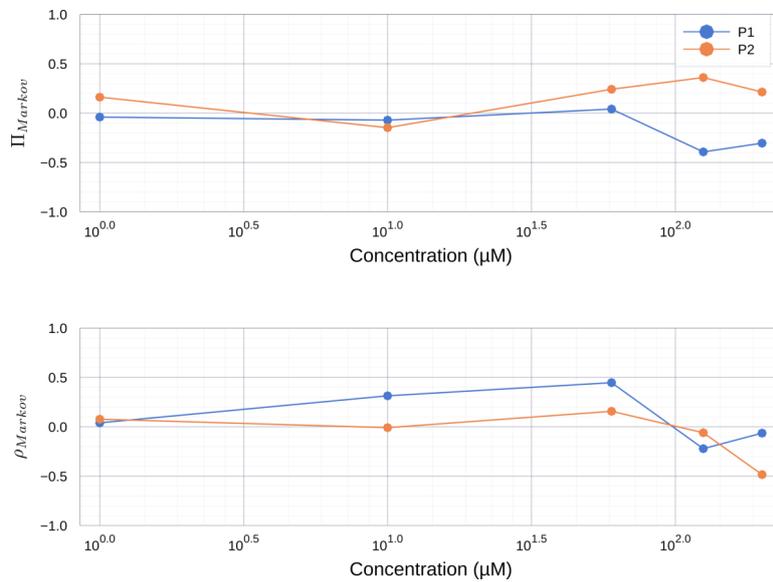


Figure I.3 – ATP: Markov chain analysis for juveniles **Top:** Mean Markov-based preference index (equation G.6). **Bottom:** Mean Markov ratio exploration-exploitation (equation G.7).

Appendix J

Preference index distributions

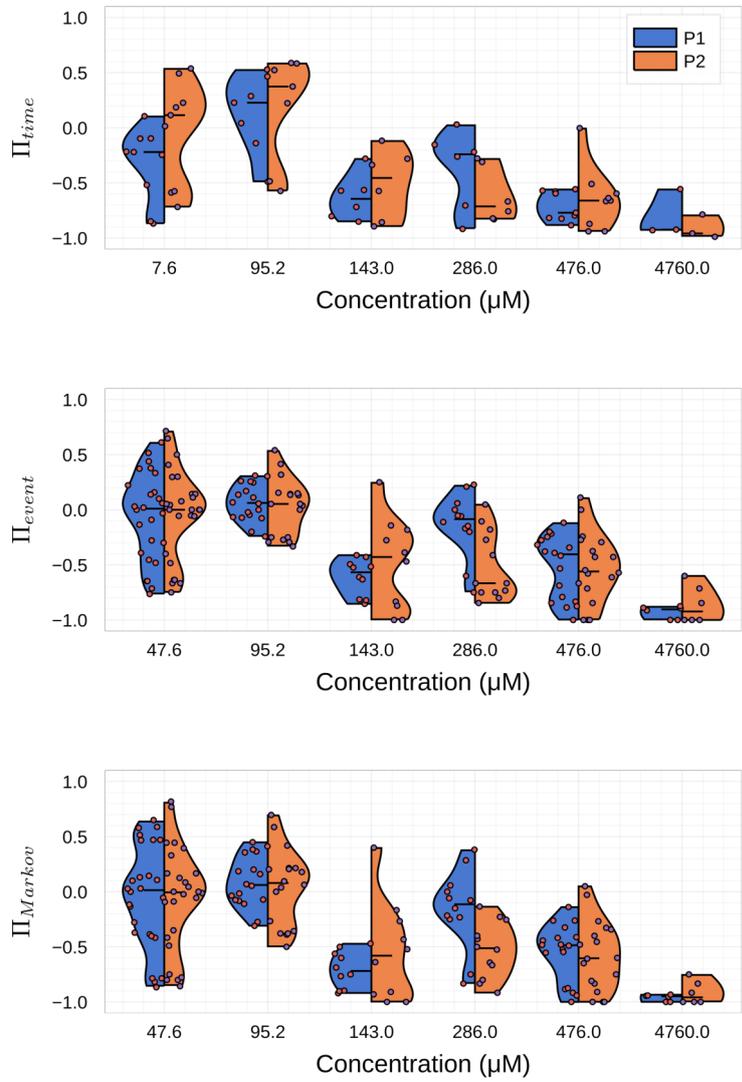


Figure J.1 – Citric acid: preference index for juveniles Event-based preference index Π_{time} , one point is one fish (top). Event-based preference index Π_{event} , one point is one fish analyzed by one person (middle). Event-based preference index Π_{Markov} , one point is one fish analyzed by one person (bottom).

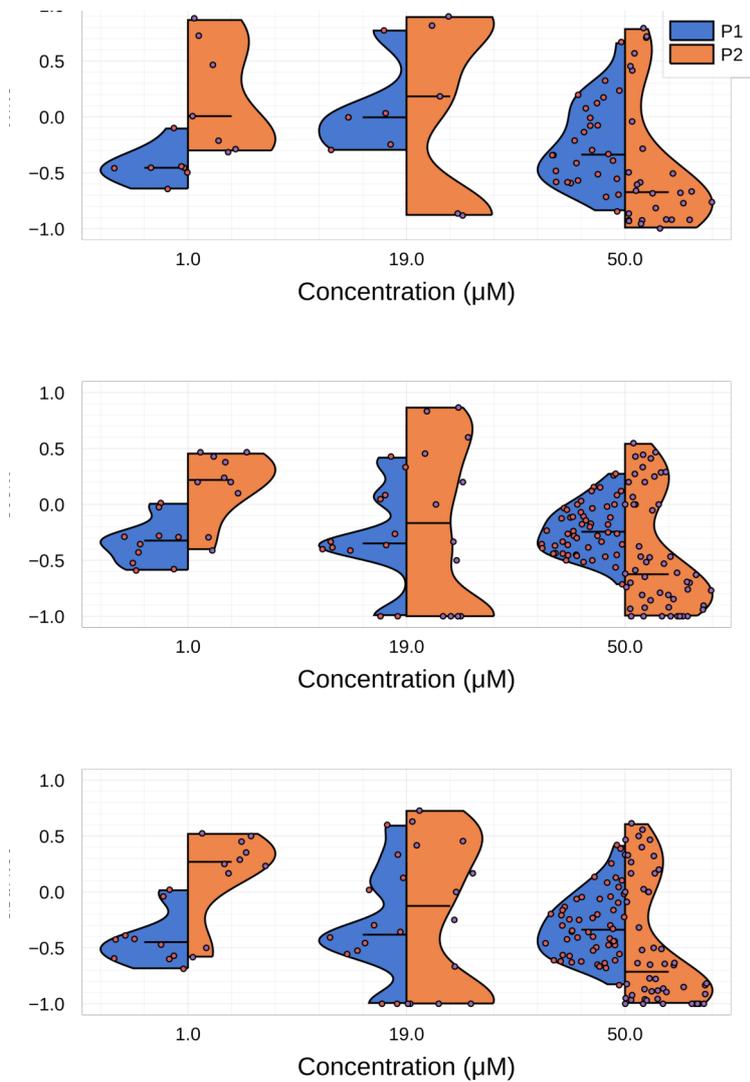


Figure J.2 – Adenosine: preference index for juveniles Event-based preference index Π_{time} , one point is one fish (top). Event-based preference index Π_{event} , one point is one fish analyzed by one person (middle). Event-based preference index Π_{Markov} , one point is one fish analyzed by one person (bottom).

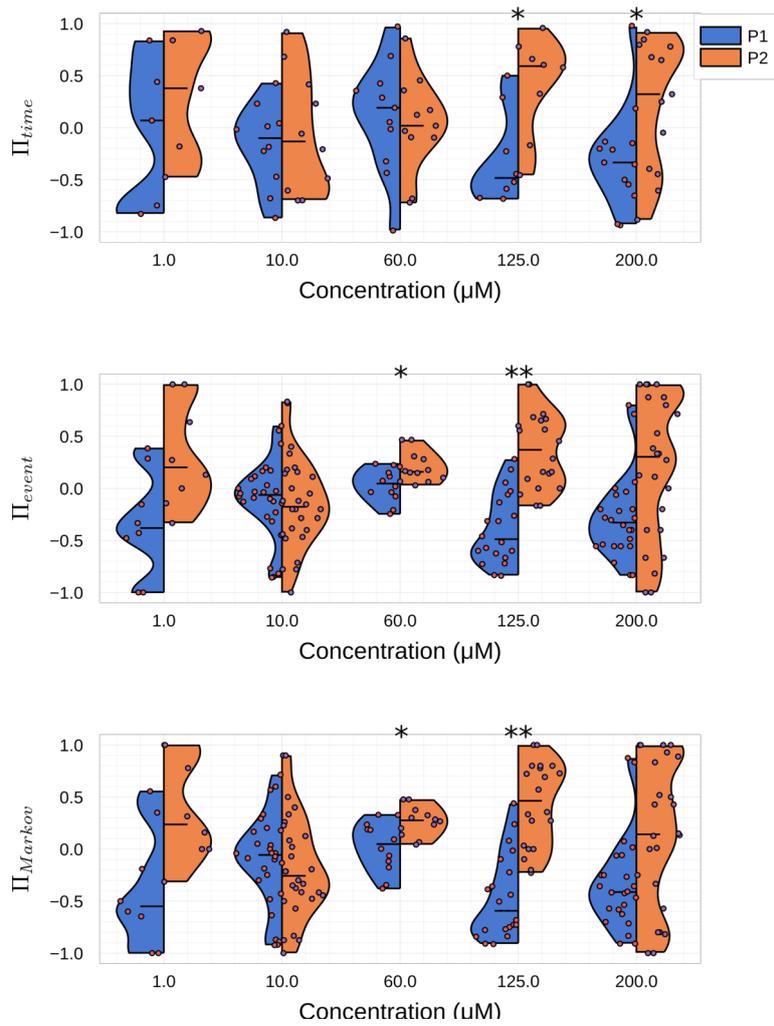


Figure J.3 – ATP: preference index for juveniles Event-based preference index Π_{time} , one point is one fish (top). Event-based preference index Π_{event} , one point is one fish analyzed by one person (middle). Event-based preference index Π_{Markov} , one point is one fish analyzed by one person (bottom). * $P < 0.05$, ** $P < 0.01$

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.
- [7] J. Jeong, N. J. Frohberg, E. Zhou, T. Sulchek, and P. Qiu, "Accurately tracking single-cell movement trajectories in microfluidic cell sorting devices," *PloS one*, vol. 13, no. 2, p. e0192463, 2018.

- [8] A. Bricard, J.-B. Caussin, N. Desreumaux, O. Dauchot, and D. Bartolo, "Emergence of macroscopic directed motion in populations of motile colloids," *Nature*, vol. 503, no. 7474, pp. 95–98, 2013.
- [9] I. Ali and M. N. Dailey, "Multiple human tracking in high-density crowds," *Image and vision computing*, vol. 30, no. 12, pp. 966–977, 2012.
- [10] A. Treptow and A. Zell, "Real-time object tracking for soccer-robots without color information," *Robotics and Autonomous Systems*, vol. 48, no. 1, pp. 41–48, 2004.
- [11] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, pp. 13–es, 2006.
- [12] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [13] Z.-M. Qian, S. H. Wang, X. E. Cheng, and Y. Q. Chen, "An effective and robust method for tracking multiple fish in video image based on fish head detection," *BMC bioinformatics*, vol. 17, no. 1, p. 251, 2016.
- [14] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. De Polavieja, "idtracker: tracking individuals in a group by automatic identification of unmarked animals," *Nature methods*, vol. 11, no. 7, pp. 743–748, 2014.
- [15] Y.-X. Bai, S.-H. Zhang, Z. Fan, X.-Y. Liu, X. Zhao, X.-Z. Feng, and M.-Z. Sun, "Automatic multiple zebrafish tracking based on improved hog features," *Scientific reports*, vol. 8, no. 1, pp. 1–14, 2018.
- [16] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "Deeplabcut: markerless pose estimation of user-defined body parts with deep learning," *Nature neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018.
- [17] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. Heras, and G. G. de Polavieja, "Idtracker. ai: tracking all individuals in small or large collectives of unmarked animals," *Nature methods*, vol. 16, no. 2, pp. 179–182, 2019.

- [18] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, P. L. Andersson, and M. Andersson, “Toxtrac: a fast and robust software for tracking organisms,” *Methods in Ecology and Evolution*, vol. 9, no. 3, pp. 460–464, 2018.
- [19] A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, and M. Andersson, “Toxid: an efficient algorithm to solve occlusions when tracking multiple animals,” *Scientific reports*, vol. 7, no. 1, pp. 1–8, 2017.
- [20] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, “Deepercut: A deeper, stronger, and faster multi-person pose estimation model,” in *European Conference on Computer Vision*, pp. 34–50, Springer, 2016.
- [21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [22] T. Nath, A. Mathis, A. C. Chen, A. Patel, M. Bethge, and M. W. Mathis, “Using deeplabcut for 3d markerless pose estimation across species and behaviors,” *Nature protocols*, vol. 14, no. 7, pp. 2152–2176, 2019.
- [23] A. Mathis and R. Warren, “On the inference speed and video-compression robustness of deeplabcut,” *BioRxiv*, p. 457242, 2018.
- [24] B. Gallois and R. Candelier, “Fasttrack: an open-source software for tracking varying numbers of deformable objects,” *PLOS Computational Biology*, vol. 17, no. 2, p. e1008697, 2021.
- [25] S. M. Easterbrook, “Open code for open science?,” *Nature Geoscience*, vol. 7, no. 11, pp. 779–781, 2014.
- [26] TheQtCompany, *The Qt Framework*, 2020. <https://www.qt.io/>.
- [27] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.
- [28] B. Stroustrup, “A history of c++ 1979–1991,” in *History of programming languages—II*, pp. 699–769, 1996.
- [29] H. S. Stone, M. T. Orchard, E.-C. Chang, and S. A. Martucci, “A fast direct fourier-based algorithm for subpixel registration of images,”

- IEEE Transactions on geoscience and remote sensing*, vol. 39, no. 10, pp. 2235–2243, 2001.
- [30] G. D. Evangelidis and E. Z. Psarakis, “Parametric image alignment using enhanced correlation coefficient maximization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1858–1865, 2008.
- [31] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International conference on computer vision*, pp. 2564–2571, IEEE, 2011.
- [32] R. C. Bolles and M. A. Fischler, “A ransac-based approach to model fitting and its application to finding cylinders in range data,” in *IJCAI*, vol. 1981, pp. 637–643, 1981.
- [33] L. Rocha, L. Velho, and P. C. P. Carvalho, “Image moments-based structuring and tracking of objects,” in *Proceedings. XV Brazilian Symposium on Computer Graphics and Image Processing*, pp. 99–105, IEEE, 2002.
- [34] R. B., “Grundlagen für eine allgemeine theorie der functionen einer veränderlichen complexen grösse.,” 1851.
- [35] M. Shahin, M. A. Babar, and L. Zhu, “Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices,” *IEEE Access*, vol. 5, pp. 3909–3943, 2017.
- [36] A. Wikström *et al.*, “Benefits and challenges of continuous integration and delivery: A case study,” 2019.
- [37] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [38] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
- [39] J. Edmonds and R. M. Karp, “Theoretical improvements in algorithmic efficiency for network flow problems,” *Journal of the ACM (JACM)*, vol. 19, no. 2, pp. 248–264, 1972.

- [40] N. Tomizawa, "On some techniques useful for solution of transportation network problems," *Networks*, vol. 1, no. 2, pp. 173–194, 1971.
- [41] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, 1987.
- [42] L. R. Yohe and P. Brand, "Evolutionary ecology of chemosensation and its role in sensory drive," *Current zoology*, vol. 64, no. 4, pp. 525–533, 2018.
- [43] L. Bardwell, "A walk-through of the yeast mating pheromone response pathway," *Peptides*, vol. 26, no. 2, pp. 339–350, 2005.
- [44] J. G. Hildebrand and G. M. Shepherd, "Mechanisms of olfactory discrimination: converging evidence for common principles across phyla," *Annual review of neuroscience*, vol. 20, no. 1, pp. 595–631, 1997.
- [45] D. A. Yarmolinsky, C. S. Zuker, and N. J. Ryba, "Common sense about taste: from mammals to insects," *Cell*, vol. 139, no. 2, pp. 234–244, 2009.
- [46] T. J. Hara, *Fish chemoreception*, vol. 6. Springer Science & Business Media, 2012.
- [47] A. Hansen and K. Reutter, "Chemosensory systems in fish: structural, functional and ecological aspects," in *The senses of fish*, pp. 55–89, Springer, 2004.
- [48] T. Panier, S. Romano, R. Olive, T. Pietri, G. Sumbre, R. Candelier, and G. Debrégeas, "Fast functional imaging of multiple brain regions in intact zebrafish larvae using selective plane illumination microscopy," *Frontiers in neural circuits*, vol. 7, p. 65, 2013.
- [49] I. H. Bianco, A. R. Kampff, and F. Engert, "Prey capture behavior evoked by simple visual stimuli in larval zebrafish," *Frontiers in systems neuroscience*, vol. 5, p. 101, 2011.
- [50] E. A. Naumann, J. E. Fitzgerald, T. W. Dunn, J. Rihel, H. Sompolinsky, and F. Engert, "From whole-brain data to functional circuit models: the zebrafish optomotor response," *Cell*, vol. 167, no. 4, pp. 947–960, 2016.

- [51] S. Wolf, A. M. Dubreuil, T. Bertoni, U. L. Böhm, V. Bormuth, R. Candelier, S. Karpenko, D. G. Hildebrand, I. H. Bianco, R. Monasson, *et al.*, “Sensorimotor computation underlying phototaxis in zebrafish,” *Nature communications*, vol. 8, no. 1, pp. 1–12, 2017.
- [52] R. Olive, S. Wolf, A. Dubreuil, V. Bormuth, G. Debrégeas, and R. Candelier, “Rheotaxis of larval zebrafish: behavioral study of a multi-sensory process,” *Frontiers in systems neuroscience*, vol. 10, p. 14, 2016.
- [53] M. Haesemeyer, A. F. Schier, and F. Engert, “Convergent temperature representations in artificial and biological neural networks,” *Neuron*, vol. 103, no. 6, pp. 1123–1134, 2019.
- [54] I. Samajdar and D. Mandal, “Histological organization and ultrastructures of the apical surface of the olfactory epithelium of a carp, *labeo bata* (hamilton),” *International Journal of Pure and Applied Zoology*, vol. 4, pp. 134–141, 2016.
- [55] M. Ichikawa and K. Ueda, “Fine structure of the olfactory epithelium in the goldfish, *carassius auratus*,” *Cell and tissue research*, vol. 183, no. 4, pp. 445–455, 1977.
- [56] A. Hansen and B. S. Zielinski, “Diversity in the olfactory epithelium of bony fishes: development, lamellar arrangement, sensory neuron cell types and transduction components,” *Journal of neurocytology*, vol. 34, no. 3-5, pp. 183–208, 2005.
- [57] G. Ahuja, S. B. Nia, V. Zapilko, V. Shiriagin, D. Kowatschew, Y. Oka, and S. I. Korsching, “Kappe neurons, a novel population of olfactory sensory neurons,” *Scientific reports*, vol. 4, no. 1, pp. 1–8, 2014.
- [58] N. Wakisaka, N. Miyasaka, T. Koide, M. Masuda, T. Hiraki-Kajiyama, and Y. Yoshihara, “An adenosine receptor for olfaction in fish,” *Current Biology*, vol. 27, no. 10, pp. 1437–1447, 2017.
- [59] N. Miyasaka, K. Morimoto, T. Tsubokawa, S.-i. Higashijima, H. Okamoto, and Y. Yoshihara, “From the olfactory bulb to higher brain centers: genetic visualization of secondary olfactory pathways in zebrafish,” *Journal of Neuroscience*, vol. 29, no. 15, pp. 4756–4767, 2009.

- [60] A. A. Nikonov and J. Caprio, "Electrophysiological evidence for a chemotopy of biologically relevant odors in the olfactory bulb of the channel catfish," *Journal of neurophysiology*, vol. 86, no. 4, pp. 1869–1876, 2001.
- [61] A. A. Nikonov, T. E. Finger, and J. Caprio, "Beyond the olfactory bulb: an odotopic map in the forebrain," *Proceedings of the National Academy of Sciences*, vol. 102, no. 51, pp. 18688–18693, 2005.
- [62] A. Hansen and Z. Eckart, "The peripheral olfactory organ of the zebrafish, *danio rerio*: an ultrastructural study," *Chemical senses*, vol. 23, no. 1, pp. 39–48, 1998.
- [63] F. Kermen, L. M. Franco, C. Wyatt, and E. Yaksi, "Neural circuits mediating olfactory-driven behavior in fish," *Frontiers in neural circuits*, vol. 7, p. 62, 2013.
- [64] R. Friedrich, E. Yaksi, B. Judkewitz, and M. Wiechert, "Processing of odor representations by neuronal circuits in the olfactory bulb," *Annals of the New York Academy of Sciences*, vol. 1170, no. 1, pp. 293–297, 2009.
- [65] O. R. Braubach, A. Fine, and R. P. Croll, "Distribution and functional organization of glomeruli in the olfactory bulbs of zebrafish (*danio rerio*)," *Journal of comparative neurology*, vol. 520, no. 11, pp. 2317–2339, 2012.
- [66] S. Serizawa, K. Miyamichi, and H. Sakano, "One neuron–one receptor rule in the mouse olfactory system," *TRENDS in Genetics*, vol. 20, no. 12, pp. 648–653, 2004.
- [67] A. L. Barth, J. C. Dugas, and J. Ngai, "Noncoordinate expression of odorant receptor genes tightly linked in the zebrafish genome," *Neuron*, vol. 19, no. 2, pp. 359–369, 1997.
- [68] F. Weth, W. Nadler, and S. Korsching, "Nested expression domains for odorant receptors in zebrafish olfactory epithelium," *Proceedings of the National Academy of Sciences*, vol. 93, no. 23, pp. 13321–13326, 1996.

- [69] Y. Sato, N. Miyasaka, and Y. Yoshihara, "Hierarchical regulation of odorant receptor gene choice and subsequent axonal projection of olfactory sensory neurons in zebrafish," *Journal of Neuroscience*, vol. 27, no. 7, pp. 1606–1615, 2007.
- [70] Y. Sato, N. Miyasaka, and Y. Yoshihara, "Mutually exclusive glomerular innervation by two distinct types of olfactory sensory neurons revealed in transgenic zebrafish," *Journal of Neuroscience*, vol. 25, no. 20, pp. 4889–4897, 2005.
- [71] R. W. Friedrich and S. I. Korsching, "Chemotopic, combinatorial, and noncombinatorial odorant representations in the olfactory bulb revealed using a voltage-sensitive axon tracer," *Journal of Neuroscience*, vol. 18, no. 23, pp. 9977–9988, 1998.
- [72] S. H. Fuss and S. I. Korsching, "Odorant feature detection: activity mapping of structure response relationships in the zebrafish olfactory bulb," *Journal of Neuroscience*, vol. 21, no. 21, pp. 8396–8407, 2001.
- [73] S. Korsching, "Odor maps in the brain: spatial aspects of odor representation in sensory surface and olfactory bulb," *Cellular and Molecular Life Sciences CMLS*, vol. 58, no. 4, pp. 520–530, 2001.
- [74] A. Hansen and E. Zeiske, "Development of the olfactory organ in the zebrafish, brachydanio rerio," *Journal of Comparative Neurology*, vol. 333, no. 2, pp. 289–300, 1993.
- [75] N. Miyasaka, A. A. Wanner, J. Li, J. Mack-Bucher, C. Genoud, Y. Yoshihara, and R. W. Friedrich, "Functional development of the olfactory system in zebrafish," *Mechanisms of development*, vol. 130, no. 6-8, pp. 336–346, 2013.
- [76] B. Kapoor, H. E. Evans, and E. Pevzner, "The gustatory system in fish," in *Advances in marine biology*, vol. 13, pp. 53–108, Elsevier, 1976.
- [77] L. Fishelson, Y. Delarea, and A. Zverdling, "Taste bud form and distribution on lips and in the oropharyngeal cavity of cardinal fish species (apogonidae, teleostei), with remarks on their dentition," *Journal of Morphology*, vol. 259, no. 3, pp. 316–327, 2004.

- [78] K. Reutter, F. Boudriot, and M. Witt, "Heterogeneity of fish taste bud ultrastructure as demonstrated in the holosteans *Amia calva* and *Lepisosteus oculatus*," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 355, no. 1401, pp. 1225–1228, 2000.
- [79] K. Reutter, "Ultrastructure of taste buds in the Australian lungfish, *Neoceratodus forsteri* (Dipnoi)," *Chem. Senses*, vol. 16, p. 404, 1991.
- [80] K. Reutter, *Taste organ in the bullhead (Teleostei)*. Springer Science & Business Media, 2012.
- [81] A. O. Kasumyan, "The taste system in fishes and the effects of environmental variables," *Journal of Fish Biology*, vol. 95, no. 1, pp. 155–178, 2019.
- [82] J. Atema, "Structures and functions of the sense of taste in the catfish (*Ictalurus natalis*)," *Brain, Behavior and Evolution*, vol. 4, no. 4, pp. 273–294, 1971.
- [83] A. Hansen, K. Reutter, and E. Zeiske, "Taste bud development in the zebrafish, *Danio rerio*," *Developmental dynamics: an official publication of the American Association of Anatomists*, vol. 223, no. 4, pp. 483–496, 2002.
- [84] Y. Ohkubo, M. Masubuchi, K. Fujioka, Y. Tomita, T. Matsushita, K. Ohsuga, and T. Marui, "Distribution and morphological features of taste buds in the zebrafish, *Danio rerio*," *Journal of Oral Biosciences*, vol. 47, no. 1, pp. 77–82, 2005.
- [85] J. Yáñez, Y. Souto, L. Piñeiro, M. Folgueira, and R. Anadón, "Gustatory and general visceral centers and their connections in the brain of adult zebrafish: a carbocyanine dye tract-tracing study," *Journal of Comparative Neurology*, vol. 525, no. 2, pp. 333–362, 2017.
- [86] K. Kotrschal, W.-D. Krautgartner, and A. Hansen, "Ontogeny of the solitary chemosensory cells in the zebrafish, *Danio rerio*," *Chemical Senses*, vol. 22, no. 2, pp. 111–118, 1997.
- [87] O. B. Stabell, "Olfactory control of homing behaviour in salmonids," in *Fish chemoreception*, pp. 249–270, Springer, 1992.

- [88] D. Hasler Arthur and A. T. Scholz, "Olfactory imprinting and homing in salmon. investigations in the mechanism of the imprinting process," 1983.
- [89] D. Pavlov, A. Kasumyan, *et al.*, "Sensory principles of the feeding behavior of fishes," *J. Ichthyol*, vol. 30, no. 6, pp. 77–93, 1990.
- [90] J. Atema, "Chemical senses, chemical signals and feeding behavior in fishes," *Fish Behaviour and its Use in the Capture and Culture of Fishes*, pp. 57–101, 1980.
- [91] W. Bateson, "The sense organs and perceptions of fishes; with remarks on the supply of bait," *Journal of the Marine Biological Association of the United Kingdom*, vol. 1, no. 3, pp. 225–256, 1890.
- [92] W. N. TAVolga, "Visual, chemical and sound stimuli as cues in the sex discriminatory behavior of the gobiid fish bathygobius soporator," *Zoologica*, vol. 41, pp. 49–64, 1956.
- [93] J. De Bruin, M. VAN DER MECHÉ-JACOBI, J. Segaar, and A. Van Der Meche, "Influence of chemical receptivity on reproductive behaviour of the male three-spined stickleback (*Gasterosteus aculeatus* L.)," *Behaviour*, vol. 86, no. 1-2, pp. 100–165, 1983.
- [94] K. v. Frisch, "Über einen schreckstoff der fischhaut und seine biologische bedeutung," *Zeitschrift für vergleichende Physiologie*, vol. 29, no. 1-2, pp. 46–145, 1942.
- [95] N. Speedie and R. Gerlai, "Alarm substance induced behavioral responses in zebrafish (*Danio rerio*)," *Behavioural brain research*, vol. 188, no. 1, pp. 168–177, 2008.
- [96] K. Doving and S. Lastein, "The alarm reaction in fishes-odorants, modulations of responses, neural pathways," *Annals of the New York Academy of Sciences*, vol. 1170, no. 1, pp. 413–423, 2009.
- [97] S. Cassar, I. Adatto, J. L. Freeman, J. T. Gamse, I. Iturria, C. Lawrence, A. Muriana, R. T. Peterson, S. Van Cruchten, and L. I. Zon, "Use of zebrafish in drug discovery toxicology," *Chemical research in toxicology*, vol. 33, no. 1, pp. 95–118, 2019.

- [98] E. W. Klee, H. Schneider, K. J. Clark, M. A. Cousin, J. O. Ebbert, W. M. Hooten, V. M. Karpayak, D. O. Warner, and S. C. Ekker, "Zebrafish: a model for the study of addiction genetics," *Human genetics*, vol. 131, no. 6, pp. 977–1008, 2012.
- [99] Y.-J. Dai, Y.-F. Jia, N. Chen, W.-P. Bian, Q.-K. Li, Y.-B. Ma, Y.-L. Chen, and D.-S. Pei, "Zebrafish as a model system to study toxicology," *Environmental toxicology and chemistry*, vol. 33, no. 1, pp. 11–17, 2014.
- [100] I. P. Pavlov, "The experimental psychology and psychopathology of animals," in *14th International Medical Congress, Madrid, Spain*, 1903.
- [101] P. Mathur, B. Lau, and S. Guo, "Conditioned place preference behavior in zebrafish," *Nature protocols*, vol. 6, no. 3, pp. 338–345, 2011.
- [102] R. Blaser and D. Vira, "Experiments on learning in zebrafish (*danio rerio*): a promising model of neurocognitive function," *Neuroscience & Biobehavioral Reviews*, vol. 42, pp. 224–231, 2014.
- [103] A. D. Collier and D. J. Echevarria, "The utility of the zebrafish model in conditioned place preference to assess the rewarding effects of drugs," *Behavioural pharmacology*, vol. 24, no. 5 and 6, pp. 375–383, 2013.
- [104] T. M. Tzschentke, "Review on cpp: Measuring reward with the conditioned place preference (cpp) paradigm: update of the last decade," *Addiction biology*, vol. 12, no. 3-4, pp. 227–462, 2007.
- [105] T. Darland and J. E. Dowling, "Behavioral screening for cocaine sensitivity in mutagenized zebrafish," *Proceedings of the National Academy of Sciences*, vol. 98, no. 20, pp. 11691–11696, 2001.
- [106] P. Mathur, M. A. Berberoglu, and S. Guo, "Preference for ethanol in zebrafish following a single exposure," *Behavioural brain research*, vol. 217, no. 1, pp. 128–133, 2011.
- [107] J. Ninkovic, A. Folchert, Y. V. Makhankov, S. C. Neuhauss, I. Sillaber, U. Straehle, and L. Bally-Cuif, "Genetic identification of ache as a positive modulator of addiction to the psychostimulant d-amphetamine

- in zebrafish," *Journal of neurobiology*, vol. 66, no. 5, pp. 463–475, 2006.
- [108] J. Ninkovic and L. Bally-Cuif, "The zebrafish as a model system for assessing the reinforcing properties of drugs of abuse," *Methods*, vol. 39, no. 3, pp. 262–274, 2006.
- [109] D. Braida, V. Limonta, S. Pegorini, A. Zani, C. Guerini-Rocco, E. Gori, and M. Sala, "Hallucinatory and rewarding effect of salvinorin a in zebrafish: κ -opioid and cb 1-cannabinoid receptor involvement," *Psychopharmacology*, vol. 190, no. 4, pp. 441–448, 2007.
- [110] X. Kedikian, M. P. Faillace, and R. Bernabeu, "Behavioral and molecular analysis of nicotine-conditioned place preference in zebrafish," *PLoS One*, vol. 8, no. 7, p. e69453, 2013.
- [111] A. J. Rennekamp and R. T. Peterson, "15 years of zebrafish chemical screening," *Current opinion in chemical biology*, vol. 24, pp. 58–70, 2015.
- [112] Y. Zhou, R. T. Cattley, C. L. Cario, Q. Bai, and E. A. Burton, "Quantification of larval zebrafish motor function in multiwell plates using open-source matlab applications," *Nature protocols*, vol. 9, no. 7, pp. 1533–1548, 2014.
- [113] V. Sallinen, V. Torkko, M. Sundvik, I. Reenilä, D. Khrustalyov, J. Kaslin, and P. Panula, "Mptp and mpp+ target specific aminergic cell populations in larval zebrafish," *Journal of neurochemistry*, vol. 108, no. 3, pp. 719–731, 2009.
- [114] J. Rihel, D. A. Prober, A. Arvanites, K. Lam, S. Zimmerman, S. Jang, S. J. Haggarty, D. Kokel, L. L. Rubin, R. T. Peterson, *et al.*, "Zebrafish behavioral profiling links drugs to biological targets and rest/wake regulation," *Science*, vol. 327, no. 5963, pp. 348–351, 2010.
- [115] D. Kokel, J. Bryan, C. Laggner, R. White, C. Y. J. Cheung, R. Mateus, D. Healey, S. Kim, A. A. Werdich, S. J. Haggarty, *et al.*, "Rapid behavior-based identification of neuroactive small molecules in the zebrafish," *Nature chemical biology*, vol. 6, no. 3, pp. 231–237, 2010.

- [116] M. J. Winter, W. S. Redfern, A. J. Hayfield, S. F. Owen, J.-P. Valentin, and T. H. Hutchinson, "Validation of a larval zebrafish locomotor assay for assessing the seizure liability of early-stage development drugs," *Journal of pharmacological and toxicological methods*, vol. 57, no. 3, pp. 176–187, 2008.
- [117] 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 tool to detect rescue of parkinsonism phenotypes in zebrafish models," *Neurobiology of disease*, vol. 44, no. 1, pp. 9–18, 2011.
- [118] Q. Shen, L. Truong, M. T. Simonich, C. Huang, R. L. Tanguay, and Q. Dong, "Rapid well-plate assays for motor and social behaviors in larval zebrafish," *Behavioural Brain Research*, p. 112625, 2020.
- [119] S. Schnörr, P. Steenbergen, M. Richardson, and D. Champagne, "Measuring thigmotaxis in larval zebrafish," *Behavioural brain research*, vol. 228, no. 2, pp. 367–374, 2012.
- [120] S. D. Pelkowski, M. Kapoor, H. A. Richendrfer, X. Wang, R. M. Colwill, and R. Creton, "A novel high-throughput imaging system for automated analyses of avoidance behavior in zebrafish larvae," *Behavioural brain research*, vol. 223, no. 1, pp. 135–144, 2011.
- [121] S. Krishnan, A. S. Mathuru, C. Kibat, M. Rahman, C. E. Lupton, J. Stewart, A. Claridge-Chang, S.-C. Yen, and S. Jesuthasan, "The right dorsal habenula limits attraction to an odor in zebrafish," *Current Biology*, vol. 24, no. 11, pp. 1167–1175, 2014.
- [122] A. Hussain, L. R. Saraiva, D. M. Ferrero, G. Ahuja, V. S. Krishna, S. D. Liberles, and S. I. Korsching, "High-affinity olfactory receptor for the death-associated odor cadaverine," *Proceedings of the National Academy of Sciences*, vol. 110, no. 48, pp. 19579–19584, 2013.
- [123] F. Kermen, L. Darnet, C. Wiest, F. Palumbo, J. Bechert, O. Uslu, and E. Yaksi, "Stimulus-specific behavioral responses of zebrafish to a large range of odors exhibit individual variability," *BMC biology*, vol. 18, no. 1, pp. 1–16, 2020.

- [124] G. D. Readman, S. F. Owen, J. C. Murrell, and T. G. Knowles, "Do fish perceive anaesthetics as aversive?," *PLoS One*, vol. 8, no. 9, p. e73773, 2013.
- [125] M. S. Abreu, A. C. V. Giacomini, D. Gusso, J. G. Rosa, G. Koakoski, F. Kalichak, R. Idalêncio, T. A. Oliveira, H. H. Barcellos, C. D. Bonan, *et al.*, "Acute exposure to waterborne psychoactive drugs attract zebrafish," *Comparative Biochemistry and Physiology Part C: Toxicology & Pharmacology*, vol. 179, pp. 37–43, 2016.
- [126] M. S. Abreu, A. C. V. Giacomini, D. Gusso, G. Koakoski, T. A. Oliveira, A. Marqueze, R. E. Barreto, and L. J. Barcellos, "Behavioral responses of zebrafish depend on the type of threatening chemical cues," *Journal of Comparative Physiology A*, vol. 202, no. 12, pp. 895–901, 2016.
- [127] K. T. L. Trinh, Q. N. Pham, and N. Y. Lee, "Clog-free and reliable solvent bonding of poly (methyl methacrylate) microdevice mediated by eco-friendly acetic acid at room temperature and its application for polymerase chain reaction and human cell culture," *Sensors and Actuators B: Chemical*, vol. 282, pp. 1008–1017, 2019.
- [128] S. Ali, H. G. Van Mil, and M. K. Richardson, "Large-scale assessment of the zebrafish embryo as a possible predictive model in toxicity testing," *PloS one*, vol. 6, no. 6, p. e21076, 2011.
- [129] O. Bénichou, M. Coppey, M. Moreau, P. Suet, and R. Voituriez, "Averaged residence times of stochastic motions in bounded domains," *EPL (Europhysics Letters)*, vol. 70, no. 1, p. 42, 2005.
- [130] A. O. Kasumyan and K. B. DÖving, "Taste preferences in fishes," *Fish and fisheries*, vol. 4, no. 4, pp. 289–347, 2003.