

**Image analysis platforms for
exploring genetic and neuronal
mechanisms regulating animal
behavior**

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Albina Asadulina, M.Sc.
aus Mailuusuu, Kirgisische Republik

Tübingen
2015

Tag der mündlichen Qualifikation:

27.04.2015

Dekan:

Prof. Dr. Wolfgang Rosenstiel

1. Berichterstatter:

Dr. Gáspár Jékely

2. Berichterstatter:

Prof. Dr. Rolf Reuter

Acknowledgements

Foremost, I would like to express my special appreciation and thanks to my advisor Dr. Gáspár Jékely for his patience, motivation and availability for discussions. Dr. Jékely has been a tremendous mentor for me encouraging my research and allowing me to grow as a research scientist.

I cordially thank the present and the former members of the AG-Jékely group, especially Nadine Randel, Aurora Panzera, Markus Conzelmann, Elizabeth Williams, Thomas Holder, Reza Shahidi, Luis Bezares, Csaba Veraszto, Jan Schlüsener for providing me with experimental data and for the fruitful discussions and help. I will keep pleasant memories of group meeting and retreats, birthday cakes and ice cream sessions on the roof terrace, dinners at the Indian restaurant, countless weekend trips and many other exciting events.

My sincere thanks also go to Prof. Dr. Karsten Borgwardt for the helpful discussions and guidance. I am grateful for the opportunity to join the very informative reading group meetings and having discussions with the members of Karsten's group.

I am very thankful to Prof. Dr. Rolf Reuter, who agreed to be part of my PhD defense committee. I want to express my gratitude to my TAC committee members Remco Sprangers and Birte Höcker for the helpful comments and advice.

During my PhD studies, I have benefited from the help of administrative and technical staff members. I especially thank Joachim Sieler and Andre Noll for the technical support and the excellent IT humor. Being a student at this institute has been a wonderful opportunity for me to learn about the cutting-edge research in many fields of biology and computer science. I greatly benefitted from the Colloquium talks, visitor presentations and tutorials organized on the campus. I really appreciate the work done by the PhD program coordinators Dagmar Sigurdardottir and Sarah Danes for advising me on with administrative issues and organizing many educational events.

I am also indebted to the developers of free software for providing the numerous tools and systems I employed for the current project. I especially would like to thank ImageJ, Fiji and Blender communities for the provided tools, support and recommendations.

I also thank Jacobo Jimenez supporting me throughout my project, for his encouragements and support in the difficult moments.

I thank Markus Conzemann and Nadine Randel who helped me with the German abstract. I thank Elizabeth Williams, Reza Shahidi, Fabio Zanini, Philipp Bauknecht, Jacobo Jimenez and Damian Roqueiro, who proofread this thesis.

Abstract

An important aim of neuroscience is to understand how gene interactions and neuronal networks regulate animal behavior. The larvae of the marine annelid *Platynereis dumerilii* provide a convenient system for such integrative studies. These larvae exhibit a wide range of behaviors, including phototaxis, chemotaxis and gravitaxis and at the same time exhibit relatively simple nervous system organization. Due to its small size and transparent body, the *Platynereis* larva is compatible with whole-body light microscopic imaging following tissue staining protocols. It is also suitable for serial electron microscopic imaging and subsequent neuronal connectome reconstruction. Despite advances in imaging techniques, automated computational tools for large data analysis are not well-established in *Platynereis*. In the current work, I developed image analysis software for exploring genetic and nervous system mechanisms modulating *Platynereis* behavior.

Exploring gene expression patterns

Current labeling and imaging techniques restrict the number of gene expression patterns that can be labelled and visualized in a single specimen, which hinders the study of behaviors driven by multi-molecular interactions. To address this problem, I employed image registration to generate a gene expression atlas that integrates gene expression information from multiple specimens in a common reference space. The gene expression atlas was used to investigate mechanisms regulating larval locomotion, settlement and phototaxis in *Platynereis*. The atlas can assist in the identification of inter-individual and inter-species variations in gene expression. To provide a representation convenient for exploring gene expression patterns, I created a model of the atlas using 3D graphics software, which enabled convenient data visualization and efficient data storage and sharing.

Exploring neuronal networks regulating behavior

Neuronal circuitry can be reconstructed from the images obtained from electron microscopy, which resolves very fine structures such as neuron morphology or synapses. The amount of data resulting from electron microscopy and the complexity of neuronal networks represent a significant challenge for manual analysis. To solve this problem, I developed the NeuroDetective software, which models a neuronal circuitry and analyzes the information flow within it. The software combines the advantages of 3D visualization and graph analysis software by integrating neuron morphology and spatial distribution together with synaptic connectivity.

NeuroDetective allowed studying the neuronal circuitry responsible for phototaxis in *Platynereis* larvae, revealing the connections and the neurons important for the network functionality. NeuroDetective facilitated the establishment of a relationship between the function and the structure

of the neuronal circuitry in *Platynereis* phototaxis.

Integrating gene expression patterns with neuronal connectivity

Neuronal circuitry and its associated modulating biomolecules, such as neurotransmitters and neuropeptides, are thought to be the main factors regulating animal behavior. Therefore it was important to integrate both genetic and neuronal information in order to fully understand how biomolecules in conjunction with neuronal anatomy elicit certain animal behavior. To resolve the difference in specimen preparation for gene expression versus electron microscopy preparations, I developed an image registration procedure to match the signals from these two different datasets. This method enabled the integration the spatial distribution of specific modulators into the analysis of neuronal networks, leading to an improved understanding of the genetic and neuronal mechanisms modulating behavior in *Platynereis*.

Kurzfassung

Eine wichtige Fragestellung der Neurowissenschaft besteht darin, wie das Zusammenspiel von bestimmten Genen und neuronalen Netzwerken das Verhalten von Lebewesen steuert. Dies lässt sich sehr gut an Larven des marinen Borstenwurmes *Platynereis dumerilii* untersuchen. Schon im frühen Stadium weisen sie ein großes Verhaltensspektrum auf - beispielsweise Phototaxis, Chemotaxis und Gravitaxis - und besitzen gleichzeitig ein relativ einfach aufgebautes Nervensystem. Aufgrund ihrer geringen Größe und des transparenten Körpers ist es möglich, histologische Färbetechniken auf der gesamten Larve anzuwenden und im Lichtmikroskop zu betrachten. Des Weiteren können Serienschritte angefertigt und so komplette neuronale Netzwerke rekonstruiert werden. Trotz fortschrittlicher Mikroskopietechniken ist die automatische Analyse großer Datenmengen für *Platynereis* jedoch nicht gut etabliert. In der zugrunde liegenden Doktorarbeit habe ich somit Software entwickelt, um die neuronalen Mechanismen und die beteiligten Gene zu analysieren, welche das Verhalten dieser Tiere steuern.

Die Erforschung von Genexpressionsmustern

Derzeitige Markier- und Visualisierungsmethoden schränken die Anzahl der Genexpressionsmuster ein, die gleichzeitig in einem Organismus erfasst werden können. Dies beeinträchtigt die Erforschung von Verhaltensmustern, welche durch mehrere Gene gesteuert werden. In der vorliegenden Arbeit habe ich Bildregistrierungsverfahren entwickelt, um die Expressionsmuster von verschiedenen Individuen in einem neuronalen Atlas zu vereinen. Der resultierende Genexpressionsatlas kann für die Analyse der Mechanismen genutzt werden, die der Bewegung, Ansiedlung und der Phototaxis in *Platynereis* zugrunde liegen. Zudem ermöglicht er die Identifikation von inter-individuellen und inter-artlichen Variationen der Expressionsmustern. Für eine benutzerfreundliche Darstellung der Genexpressionsmuster habe ich ein 3D Modell des Atlas gestaltet, welches eine geeignete Visualisierung und eine effiziente Speicherung der erhaltenen Daten sicherstellt.

Verhaltenssteuernde neuronalen Mechanismen

Neuronennetzwerke können aus hoch-auflösenden elektronenmikroskopischen Bildern rekonstruiert werden. Hierbei stellen jedoch die Datenmenge und die Komplexität der Netzwerke eine große Herausforderung für die Wissenschaft dar. Um dieses Problem zu lösen, habe ich NeuroDetective entwickelt: eine Software, die Neuronennetzwerke modelliert und den Informationsfluss innerhalb des Netzwerks analysiert. Sie kombiniert die Vorteile von 3D Visualisierungs- und Graphenanalysesoftware und integriert neuronale Morphologie und synaptische Verbindungen. Mithilfe dieser Software wurde das für die Phototaxis zuständige Neuronennetzwerk in *Platynereis* analysiert. NeuroDetective stellt damit die Verbindung zwischen

Funktion und Struktur des für die Phototaxis verantwortlichen neuronalen Netzwerkes dar.

Integration von Genexpressionen mit dem Neuronennetzwerk

Das Neuronennetzwerk und dessen modulierende Biomoleküle, wie Neurotransmitter und Neuropeptide, gelten als Hauptfaktoren für die Verhaltenssteuerung in Tieren. Um dieses Zusammenspiel zu verstehen, war es äußerst wichtig, molekulare und strukturelle Daten zu kombinieren. Aufgrund der unterschiedlichen Präparationsmethoden für Licht- und Elektronenmikroskopie habe ich eine spezielle Bildregistrierungsmethode entwickelt, um die Daten einander anzupassen. Diese Methode ermöglicht die Erforschung der Genexpression im Kontext feiner morphologischer Details sowie die Analyse von Neuronennetzwerken unter Berücksichtigung ihrer Modulatoren.

Contents

1. Introduction	1
2. Background	3
2.1. Computational techniques	3
2.1.1. Image analysis	3
2.1.2. Image registration	11
2.2. Biological background	13
2.2.1. <i>Platynereis dumerilii</i>	13
2.2.2. Molecular techniques established in <i>Platynereis</i>	15
3. Gene expression atlas for <i>Platynereis dumerilii</i>	17
3.1. Introduction	17
3.1.1. Anatomical atlases in solving biological questions	17
3.1.2. Image registration for atlas generation	18
3.1.3. Extended atlases	19
3.1.4. Current work contribution	19
3.2. Methods	20
3.2.1. <i>In situ</i> hybridization	20
3.2.2. Immunohistochemistry	20
3.2.3. Confocal microscopy	21
3.2.4. Image registration	21
3.2.5. Template generation	23
3.2.6. Gene expression pattern registration	24
3.2.7. Multichannel visualization	24
3.2.8. Model for gene expression atlas	25
3.2.9. Mobile GeneExpressionAtlas	26
3.2.10. Online GeneExpressionAtlas	27
3.2.11. Pharmacological experiments	28
3.3. Results	30
3.3.1. Evaluation of image registration	30

3.3.2.	Average anatomy templates for <i>Platynereis dumerilii</i>	32
3.3.3.	Gene expression pattern registration	34
3.3.4.	Registration of immunostained specimens	36
3.3.5.	Linking anatomical structure and function	37
3.3.6.	Interactive gene expression atlas	39
3.3.7.	Mobile and web applications for <i>Platynereis</i> gene expression atlases . .	39
3.3.8.	Gene expression colocalization using gene expression atlas	42
3.4.	Discussion	44
3.4.1.	Image registration protocol	44
3.4.2.	Comparison with VIB, ISA and PrImR protocols	45
3.4.3.	Nuclear stains as universal reference markers	46
3.4.4.	Immunochemistry for behavior-regulating neuropeptides	47
3.4.5.	Interactive model of the gene expression atlas	47
3.4.6.	Gene expression atlas in <i>Platynereis</i> research	48
3.4.7.	Future directions	48
4.	Neuronal circuitry model for <i>Platynereis dumerilii</i>	50
4.1.	Introduction	50
4.1.1.	Nervous system reconstructions in model organisms	50
4.1.2.	Software for electron microscopy images	51
4.1.3.	A new software tool for neuronal network exploration	53
4.2.	Methods	55
4.2.1.	Electron microscopy	55
4.2.2.	Neuronal network reconstruction	55
4.2.3.	Neuronal network model	55
4.2.4.	NeuroDetective functionality	58
4.2.5.	Mobile NeuroDetective	61
4.2.6.	Online NeuroDetective	61
4.2.7.	Analysis of eye-ablation experiments	62
4.2.8.	Analysis of light-contrast experiments	62
4.3.	Results	64
4.3.1.	NeuroDetective: a software tool for exploring neuronal connectivity . .	64
4.3.2.	Model for <i>Platynereis</i> neuronal network	67
4.3.3.	Analysis of synaptic connectivity in <i>Platynereis</i>	68
4.3.4.	Centrality measure for <i>Platynereis</i> neuronal network	71
4.3.5.	Additional information for the <i>Platynereis</i> neuronal network	73
4.3.6.	Mobile and web applications for the <i>Platynereis</i> neuronal network . . .	75

4.3.7. Neuronal mechanisms of phototaxis in <i>Platynereis</i>	75
4.4. Discussion	79
4.4.1. A tool for visualization and analysis of a neuronal network	79
4.4.2. Software for visual data analytics	81
4.4.3. Knowledge base of <i>Platynereis</i> nervous system	82
4.4.4. Understanding neuronal circuitry function	83
4.4.5. <i>Platynereis</i> neuronal circuitry for mobile and online exploration	83
4.4.6. Synaptic connectivity in other species	84
4.4.7. Future perspectives	84
4.4.8. Conclusion	85
5. Integrating light and electron microscopy datasets	87
5.1. Introduction	87
5.2. Methods	88
5.2.1. Image set from electron microscopy	88
5.2.2. Registration of the light and electron microscopy images	89
5.3. Results	90
5.3.1. Modification of the electron microscopy stack	90
5.3.2. Light microscopy signals mapped to the electron microscopy stack	92
5.4. Discussion	95
6. Discussion	97
Contributions	100
Publications	101
List of Figures	101
List of Tables	103
A. Supplementary files	105
B. Instructions	108
Bibliography	117

1. Introduction

Behavior is the observable activity of a living organism in response to internal or external stimuli [1]. Ethology, the study of animal behavior, emerged in the middle of the 20th century. The key founders of ethology were Konrad Lorenz and Niko Tinbergen, who suggested that each behavior should be characterized in terms of four levels of explanation: *causation*, *development*, *function* and *evolution* [2]. *Causation* is concerned with mechanisms underlying behavior (for example, genes or neuronal circuits) and *function* considers the objective of behavior. *Development* addresses the modification of behavior throughout animal life, whereas *evolution* considers how behavior has changed in the course of evolution.

Research on behavior is important to various scientific disciplines, such as ecology, neuroscience or developmental biology. Animal behavior research is essential for protecting endangered species. Understanding organism's reproductive behavior and interactions with other species leads to the development of more efficient strategies for conservation [3, 4]. Understanding animals' interaction with their environment (such as determining reproductive strategies in insects) is vital for guiding the development of environmentally friendly methods for pest population control [5] or disease transmitting species [6, 7, 8]. Studying behavior from an evolutionary perspective advances our knowledge of the origin, development and ancestral relationships of behaviors and reveals the importance of certain behaviors for survival [9, 10]. In neuroscience, behavioral analysis is an essential step in establishing the relationship between the brain and its function [11, 12]. Discoveries in brain-behavior relationship led to the development of bio-inspired algorithms and technologies, for example, artificial neural networks [13] or animal robots [14, 15].

The procedures originally employed for studying animal behavior (such as observation and description) have recently been enriched by sophisticated experimental manipulations and data analysis techniques. Various electronic devices and detectors have been invented to access animal habitats unreachable for humans (such as highlands or deep seas) or to capture signals inaccessible for human perception (such as ultrasound, infrared light or chemicals).

Due to the observational nature of behavior, visual recording is frequently applied in behavior research. Subsequent analysis of the acquired experimental data can be challenging or impossible for humans due to its complex nature, multidimensionality or large volume. Various image-analysis techniques have been applied to extract meaningful information efficiently and

in an unbiased manner or to represent data in a convenient form for analysis.

Image analysis has been widely employed for studying behavior in *C.elegans* [16, 17, 18, 19], *Drosophila* [20, 21, 22, 23, 24, 25, 26, 27, 28], mouse [20, 29], zebrafish [30, 31] and the marine invertebrates *Platynereis* [32, 33, 34] and *Crepidula fornicata* [35]. Tools for behavioral analysis provided direct statistical characteristics of animal behavior such as locomotion speed for individual animals [23, 31, 35] or distance between two animals for social interaction studies [29, 31]. In some cases, behavior was *a priori* classified for a subsequent characterization. To classify animal behavior, rule-based approaches [16, 23, 31] or machine learning techniques have been frequently applied [21, 22]. GentleBoost [20], Classification and Regression Trees (CART) [18, 19], Bayesian framework [27] and support vector machines [28] were among the successful machine learning techniques.

Besides post-recording analysis, some tools characterized behaviors in the course of the experiments [16, 27, 31], thereby enabling adjustment of the experimental settings based on preceding experimental outcomes. Most of the considered behavioral analysis tools operated on 2D videos, however, in some cases, it was important to track animal behavior in three dimensions. Thus, synchronized multicameras were applied to obtain the 3D coordinates of an animal for subsequent characterization [27, 28].

The tools summarized above are mostly concerned with the *causation* level of ethology, as they aim to understand intrinsic mechanisms modulating behavior, such as the role of neuronal nicotinic receptors in the establishment of social interactions [22], the role of spinal projection neurons in zebrafish turning behaviors [30] or the role of the neuropeptides in *Platynereis* vertical swimming [32] and settlement [33]. Some tools focus on genetic and gender factors yielding specific variations in behavior [19, 20, 21]. Other tools deal with the change of behavior in the presence of varying stimuli, which could be chemical [16, 26], mechanistic [16, 25], visual [25, 26, 27] or thermal [17, 23, 26].

Like previous data analysis tools, the current project is also concerned with the *causation* level of ethology, providing a platform for characterizing behavior in the marine annelid *Platynereis dumerilii* and understanding the genetic and nervous system mechanisms modulating *Platynereis* larval behavior.

2. Background

2.1. Computational techniques

2.1.1. Image analysis

The goal of image analysis is to extract quantitative or qualitative information from image data [36]. Due to progress in imaging technology, the amount of digital image data is steadily growing, therefore there is a need to develop computational techniques for analyzing large amounts of images, perform complex operations on them and extract quantitative information. The overview of image analysis is largely based on ref. [37]. Image analysis overlaps with the fields of computer vision, pattern recognition, and signal processing and employs techniques in mathematics, pattern recognition, artificial intelligence, robotics, computer science, electronics, and other fields.

The image analysis techniques can be divided into low-level image processing and high-level image understanding. Low-level methods usually focus on image processing, such as compression, noise filtering, intensity adjustment or edge detection. They take an image as input and give a transformed image as output. High-level methods aim at extracting real-world concepts from images and subsequently characterizing those concepts, for example, defining object size, shape, relative composition, inter-object distance, etc. They take an image as input and transform it into another representation (number, model, etc.).

Image pre-processing

Image pre-processing is the transformation of the image with the aim of suppressing unwanted signal and enhancing the desired features for analysis. There are four types of pre-processing based on the size of the neighborhood defined for a pixel: **(1) brightness**, **(2) geometric transformation**, **(3) filtering** and **(4) image restoration**.

(1) Brightness transformation is concerned with modifying pixel intensity in a systematic manner. The transformation can be a function of position, which can be used to compensate for an uneven illumination, or a function of original intensity, which can be used for contrast enhancement.

(2) Geometric transformation is a spatial transformation of the image coordinates, which aims to compensate for image distortions occurring when the image is captured (e.g. due to lens distortion or camera orientation). It is widely used in image registration. Geometric transformations can be classified as (A) *rigid*, (B) *affine* and (C) *deformable*.

(A) *Rigid geometric transformation* preserves image shape and size. It includes translations, rotations and reflections. Transformations can be represented by a matrix multiplication $y = Mx$, where x is the original coordinate vector, M is the transformation matrix and y is the transformed coordinate vector:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}$$

in three-dimensional space. Translation can be represented as:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix},$$

where q_1, q_2, q_3 are translations in x, y, z directions respectively. Rotation of q_1 degrees about the x axis can be described as:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q_1) & \sin(q_1) & 0 \\ 0 & -\sin(q_1) & \cos(q_1) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}$$

and rotation of q_2 and q_3 about y and z axes respectively:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(q_2) & 0 & \sin(q_2) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(q_2) & 0 & \cos(q_2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} \text{ and } \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(q_3) & \sin(q_3) & 0 & 0 \\ -\sin(q_3) & \cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}$$

(B) *Affine geometric transformation* preserves ratios of distances and collinearity (i.e. all points lying on a line before transformation, still lie on a line after transformation). It includes rigid transformations, scaling and shearing. Scaling can be described as:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix},$$

where q_1, q_2, q_3 are scaling factors in x, y, z directions respectively. Shearing by parameters q_1, q_2, q_3 can be performed as:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & q_1 & q_2 & 0 \\ 0 & 1 & q_3 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}$$

Combined transforms can be constructed by multiplying together transformation matrices.

(C) *Deformable transform* defines a transformation vector for each pixel. The deformation can be defined using radial basis functions, physical or deformation models.

Interpolation is used for mapping from one image space to another. After applying a geometric transformation, the obtained coordinates might lie outside the nodes of the discrete grid of the output image (i.e. non-integer coordinates). The values on the nodes of the integer grid have to be specified, which can be obtained by brightness interpolation of some neighboring non-integer samples. There are multiple interpolation methods, the most popular of which are:

- nearest neighborhood interpolation, which assigns to the point the brightness value of the nearest sample point,
- linear interpolation, which assumes that the brightness function is linear in this neighborhood,
- bicubic interpolation, which approximates brightness field by a bicubic polynomial function.

(3) Filtering modifies pixel intensities based on neighboring pixels. Filtering can be classified according to the goal of processing: smoothing, which aims to suppress small fluctuations in the image, and gradient operators, which are based on local image derivatives. Image smoothing techniques suppress image noise using redundancy in the image data. Smoothing suppresses high frequencies in the Fourier transform domain. Numerous methods are developed for image smoothing including such filters as:

- Gaussian filter, which convolves image with a Gaussian kernel of pixel radius,
- average smoothing, which replaces the current point by average intensity in its neighborhood respectively,

- median smoothing, which replaces the current point by median intensity in its neighborhood respectively.

The aim of gradient operators is to indicate the locations in the image where the image intensity function undergoes rapid changes or, in other words, where the derivatives are bigger. Gradient operators suppress low frequencies in the Fourier transform domain. Convolution, which replaces pixel values by the weighted sum of its neighbors, is one of the popular methods. The weights are determined by a kernel matrix. Laplace, Prewitt, Sobel or other kernel matrices can be used.

(4) Image restoration aims to suppress image distortion using knowledge about its nature. Distortion can have various causes: defects of imaging device optical lenses, motion of a captured object, incorrect focus and others. The objective of image restoration is to reconstruct the original image from its distorted version. Image restoration techniques can be classified as deterministic and stochastic. Deterministic methods obtain the original image from the distorted one by a transformation inverse to the degradation (e.g. knowing lens distortion function). Stochastic techniques try to find the best restoration using a particular stochastic criterion, such as least squares or other interpolation methods.

Segmentation

Image segmentation divides an image into parts that correspond to objects or areas which are important for a subsequent analysis of image data. The segmented objects or areas are homogeneous with respect to a selected property, which can be brightness, color or texture. Segmentation methods can be classified according to the dominant features they employ: **(1) thresholding**, **(2) edge-based segmentation**, **(3) region-based segmentation** and **(4) matching**.

(1) Thresholding methods determine a brightness constant or a threshold in order to segment objects from the background. Grey level thresholding is a simple but very effective segmentation method, since real-world objects are often characterized by constant reflectivity or light absorption. Thresholding is the oldest segmentation method and is still widely applied, since it is computationally cheap and fast. Thresholding methods can be categorized as follows [38]:

- Histogram shape-based thresholding is based on the shape properties of a histogram.
- Clustering-based thresholding performs any type of clustering analysis on gray-level data.
- Entropy-based thresholding exploits the entropy of the distribution of gray levels in a scene.
- Thresholding based on attribute similarity selects a threshold value based on some attribute quality or similarity measure between the original image and the binarized version of that image.

- Spatial thresholding utilizes both the gray value distribution and the pixel dependency in the neighborhood.
- Locally adaptive thresholding depends on some local statistical parameters of the pixel neighborhood.

(2) **Edge-based segmentation** relies on the previously described edge-detecting operators, which mark image locations of discontinuities in grey level, color, etc. The edges are then combined into chains that correspond to the object borders with the final goal to reach object or area segmentation.

(3) **Region growing segmentation** aims at defining image regions with homogenous properties (color, gray level, texture, etc). Region growing techniques generally perform better than other methods in noisy images where edges are extremely difficult to detect and threshold is difficult to define. Simple versions of region growing include merging, splitting, and split-and-merge approaches.

(4) **Matching** is another basic approach to segmentation, used to locate known objects in an image or to search for specific patterns. The match-based methods can be adapted to detect patterns that are rotated or scaled. For that, patterns of all possible sizes and rotations should be considered or an image should be matched with all possible geometric transforms of a single pattern.

Object recognition

Object recognition aims to assign classes to objects or regions. Frequently, machine learning is applied to recognize objects from images. Machine learning (ML) is a scientific discipline that enables computers to automatically learn from data. It takes data as input and gives as output algorithms capable of performing a desired task. Depending on the training approach they use, ML strategies can be classified as (1) **supervised**, (2) **unsupervised** and (3) **reinforcement learning**. The current overview of machine learning is based on ref. [39, 40].

(1) **Supervised learning**. Given a sample of input-output pairs, called the training set, the task is to find a function that maps any input to an output and is used to predict output for future observations. According to the output data type, supervised learning can be distinguished as regression or classification learning. In regression, the output space represents the values of continuous variables, for instance some physical measure (e.g. pressure, temperature). Linear or nonlinear regressions can be used for learning.

In classification learning, the output space represents a set of discrete classes or categories. The learning algorithm that solves the classification problem is called a classifier. (A) *Naive Bayes classifier*, (B) *Support Vector Machines* and (C) *Neural networks* are just a few examples of methods used in classification.

2. Background

(A) *Naive Bayes classifier*: Given a set of variables, $X = \{x_1, x_2, x_3, \dots, x_n\}$, the task is to construct the posterior probability for the event C_i among a set of possible outcomes $C = \{C_1, C_2, C_3, \dots, C_m\}$ using Bayes' rule:

$$p(C_j|x_1, x_2, x_3, \dots, x_n) \propto p(C_j) * p(x_1, x_2, x_3, \dots, x_n|C_j)$$

where $p(C_j|x_1, x_2, x_3, \dots, x_n)$ is the posterior probability of belonging to a class C_j . Naive Bayes assumes independence of the conditional probabilities of the independent variables, thus, the likelihood can be represented as a product of terms:

$$p(X|C_j) \propto \prod_{k=1}^n p(x_k|C_j)$$

The posterior can then be rewritten as:

$$p(C_j|X) \propto p(C_j) \prod_{k=1}^n p(x_k|C_j)$$

Using Bayes rule above, a new case X is labeled with a class C_j with the highest posterior probability.

(B) *Support Vector Machines (SVM)* perform classification tasks by constructing hyperplanes in a multidimensional parameter space that separates objects of different classes. SVM can handle multiple continuous and categorical variables and therefore supports both regression and classification tasks. An iterative training algorithm is employed to construct an optimal hyperplane, which minimizes an error function:

$$\operatorname{argmin}_{(w,b)} \frac{1}{2} \|w\|^2$$

$$\text{subject to: } y_i(wx_i + b) \geq 1 \forall i$$

where x_i represents independent variables, y_i represents class labels, w is the vector of coefficients and the expression $\frac{b}{\|w\|}$ determines the offset of the hyperplane from the origin.

(C) *Supervised Neural Networks* were inspired by the behavior of biological neurons [37]. The basic element of a neural network is a neuron, which can be represented by a linear discrimination function. A single neuron is equivalent to a linear classifier that classifies patterns into two classes. Its input is a feature vector \mathbf{x} and its output is a scalar:

$$y = f(\sum_{i=1}^n (w_i x_i + w_0))$$

where \mathbf{w} consists of weights defining the discrimination hyperplane and f is a nonlinear function, often in a threshold or sigmoid form. A neural network consists of a set of nodes (single neurons) and connections between them. The network properties are determined by its architecture, by the node behavior, the inter-node weights \mathbf{w} , the thresholds w_0 , and the nonlinear

function f . There are nodes that are directly connected to the input or output, and nodes with no direct connection to input or output (so-called interlayers, or hidden nodes). Neural networks can be divided into two main groups according to their architecture: recurrent and feed-forward networks. Recurrent networks contain connections from later neurons to earlier neurons, i.e. they contain cycles. Feed-forward networks represent acyclic directed graphs.

(2) Unsupervised learning. The input data represent only a sample of objects without associated target values. The output of the learning algorithm can be a set of clusters or a probability density of how likely it is to observe a certain object in the future. Unsupervised learning takes advantage of a structure possibly existing in the training data. Clustering algorithms form a large subclass of unsupervised learning. Given a fixed number of clusters, the algorithms aim at finding a grouping of objects such that the similarities of the objects in one cluster are much greater than the similarities among the objects from different clusters.

Clustering methods can be categorized into partitioning and hierarchical clustering. Partitioning methods directly assign data points into different clusters according to some appropriate criteria, such as similarity or probability density. (A) *K-means*, (B) *expectation-maximization* and (C) *hierarchical clustering* are popular partitioning algorithms.

(A) *K-means* is a classic clustering technique [40]. The number of clusters k has to be specified in advance. Then k points are chosen randomly as cluster centers. All instances are assigned to their closest cluster center according to the Euclidean distance metric in the parameter space. Next the center of the instances in each cluster is calculated. The calculated centroids are chosen as new center values for the respective clusters. The procedure is then repeated with the new cluster centers. Iteration continues until the same objects are assigned to the clusters in consecutive rounds.

(B) *The expectation-maximization (EM)* algorithm seeks for the maximum likelihood estimates of the parameters of an underlying probability distribution for a given data set. The aim is to maximize the posterior probability of the parameters θ given the observed data x marginalizing over hidden variables z .

The EM algorithm iteratively performs two steps: *Expectation*, which computes an expectation of the likelihood by including the hidden variables as if they were observed, and *Maximization*, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found in the E-step. The first estimate θ_0 is usually randomly initialized. The parameters found in the M-step are then used to begin the subsequent E-step, and the process is then repeated. The following is the outline of the EM algorithm:

1. Initialize θ_{old} .
2. E-Step: Estimate the distribution $\Phi(z|x, \theta_{old})$ of the unobserved z variables, conditioned on the observation, using the values from the last M-step:

2. Background

$$Q(\theta, \theta_{old}) = E[\log p(x, z | \theta) | x, \theta_{old}]$$

3. M-Step: Using the values from previous E-step, compute a new estimate for the parameters:

$$\theta = \operatorname{argmax}_{\theta} Q(\theta, \theta_{old}).$$

4. Repeat until converge.

In practice, the EM algorithm converges after few iterations, making it popular in the machine learning community.

(C) *Hierarchical clustering* sequentially builds nested clusters in the form of a dendrogram. This method can be categorized in two different groups: agglomerative and divisive. Agglomerative methods use a bottom-up approach: they start with n clusters formed by a single pattern and build the dendrogram merging them successively. Divisive methods are top-down: they start with a single cluster including all the patterns and then split a cluster into two other clusters and so on. The popular agglomerative method is the agglomerative hierarchical clustering:

1. Assign each object to a cluster.
2. Construct a matrix of pairwise distances between clusters.
3. Find the two nearest clusters S_i and S_j .
4. Merge S_i and S_j and remove them from the matrix.
5. Update the matrix by evaluating distances from the new cluster to all other clusters.
6. Repeat until all clusters are merged into a single cluster.

Several definitions of the distance metric between clusters have been proposed. The most popular are:

- Single-link clustering: the minimum distance between objects from two clusters.
- Complete-link clustering: the maximum distance between objects from two clusters.
- Average linkage clustering: weighted sum of distance between objects from two clusters.
- Centroid clustering: the distance between the centroids of two clusters.

(3) Reinforcement learning The objective is to learn how to map actions to an environment in order to maximize a given reward. It is not specified which actions to take in a given situation. Reinforcement learning algorithms gain information about the actions taken by means of some reward or punishment. One of the biggest challenges of reinforcement learning is to find a trade-off between exploration and exploitation. On one hand, the learning algorithm must choose effective actions, which have been tried out in the past, i.e. exploit its current knowledge. On the other hand, it has to choose novel actions to discover new effective actions and thus explore the state space.

Image interpretation and computer vision

The information obtained from the described image analysis methods can be further characterized or can serve as a preparation step for image interpretation, such as action detection or scene classification (e.g. understanding image semantics). Image interpretation is the main goal in computer vision, which aims to achieve visual perception similar to that of biological systems. The challenge is that biological perception systems include complex control strategies incorporating parallel processing, dynamic sensing, behavior modifications, shift of attention focus, etc. Efficient coordination strategies are critical in computer vision, since they are concerned with the appropriate sequence of processing steps: various methods of image processing, object recognition and characterization described earlier.

2.1.2. Image registration

Registration is the determination of a geometrical transformation that aligns points in one view of an object with the corresponding points in another view of that object or another object [41]. Geometrical transformation refers to a spatial transformation of the image coordinates and was described previously (cf. Image pre-processing). According to registration basis, image registration methods can be classified as **(1) point-**, **(2) surface-** and **(3) intensity-based methods**.

(1) Point-based registration requires a predetermined set of corresponding point pairs for a given pair of images. The predefined points are called fiducial points or landmarks. The registration procedure seeks for a transformation that aligns the corresponding fiducial points and will then interpolate the mapping from these points to other points in the images.

(2) Surface-based registration requires the boundary surface of an object or structure. The registration procedure determines corresponding surface regions in two objects and seeks for a transformation that aligns these regions.

(3) Intensity-based registration involves calculating a transformation between two images using the pixel or voxel intensity values. The transformation is determined by iteratively optimizing some image similarity metric calculated from image intensity values. Several similarity metrics have been implemented for measuring the matching accuracy:

- Image subtraction uses the sum of squares of intensity differences (SSD):

$$SSD = \frac{1}{N} \sum_i^N |A(i) - B(i)|^2,$$

where $A(i)$ and $B(i)$ are pixels in the images A and B respectively. The metric is zero when the images are perfectly aligned and is proportional to the registration error.

- Correlation coefficient of the pixel intensities can be applied if the intensities in images are linearly related. However, few application cases conform to this requirement.

- Ratio-image uniformity (RIU) is an intramodality metric calculated as normalized standard deviation of intensity ratios of two images:

$$RIU = \frac{\sigma_R}{\mu_R},$$

where R is image ratio $R(i) = A(i)/B(i)$, μ_R is the mean of R : $\mu_R = \frac{1}{N} \sum_i R(i)$ and σ_R is the standard deviation of R : $\sigma_R = \sqrt{\frac{1}{N} \sum_i (R(i) - \mu_R)^2}$.

- Mutual information normalizes joint entropy with respect to the partial entropies of contributing signals:

$$I(A, B) = H(A) + H(B) - H(A, B),$$

where $H(A)$ is the Shannon entropy:

$$H(A) = -\sum_i p(i) \log p(i),$$

where $p(i)$ is a probability distribution density function of the image A .

Registration transform can be determined using optimization algorithms, which search for an optimal similarity metric over a transform parameter space. Many techniques have been proposed, some of the popular ones are:

Gradient descent: given a scalar function $f(x)$ with $x \in R_n$, we want to find its minimum $\min f(x)$. The gradient descent method starts from the initial point x_0 and then iteratively takes a step along the steepest descent direction, until an optimal solution is found.

Simulated annealing models the physical process of heating a material and then slowly cooling it to decrease defects, thus minimizing the system energy [42]. The basic idea is to generate a path through the parameter space, from point to point, leading ultimately to an optimal solution. A subsequent point is chosen from the neighborhood of the previous point by a probabilistic function. Steps are not strictly required to produce improved solutions, but with certain probability lead to improvement. Simulated annealing can obtain optimal solutions without becoming trapped in a local optimal point.

Genetic algorithms are based on a natural selection process that imitates biological evolution. The algorithm iteratively modifies a population of individual solutions. At each iteration, the algorithm selects individuals from the current population based on a fitness score. The selected individuals undergo "mutation" and "recombination" to breed the next generation. The population iteratively "evolves" toward an optimal solution.

2.2. Biological background

2.2.1. *Platynereis dumerilii*

The marine annelid *Platynereis dumerilii* is a model organism in evolutionary developmental biology, ecology and neurobiology. Due to the relatively slow rates of its evolution, *Platynereis* possesses cell types that share common ancestry with cell types present in vertebrates and other invertebrates [43]. Many genes and cell types in *Platynereis* are evolutionarily ancestral and representative of those in the bilaterian ancestor. The *Platynereis* genome shares more features with vertebrate genomes than insect or nematode genomes do [44].

Platynereis has a complex life cycle [45]. Starting as an embryo in the egg jelly, it transforms

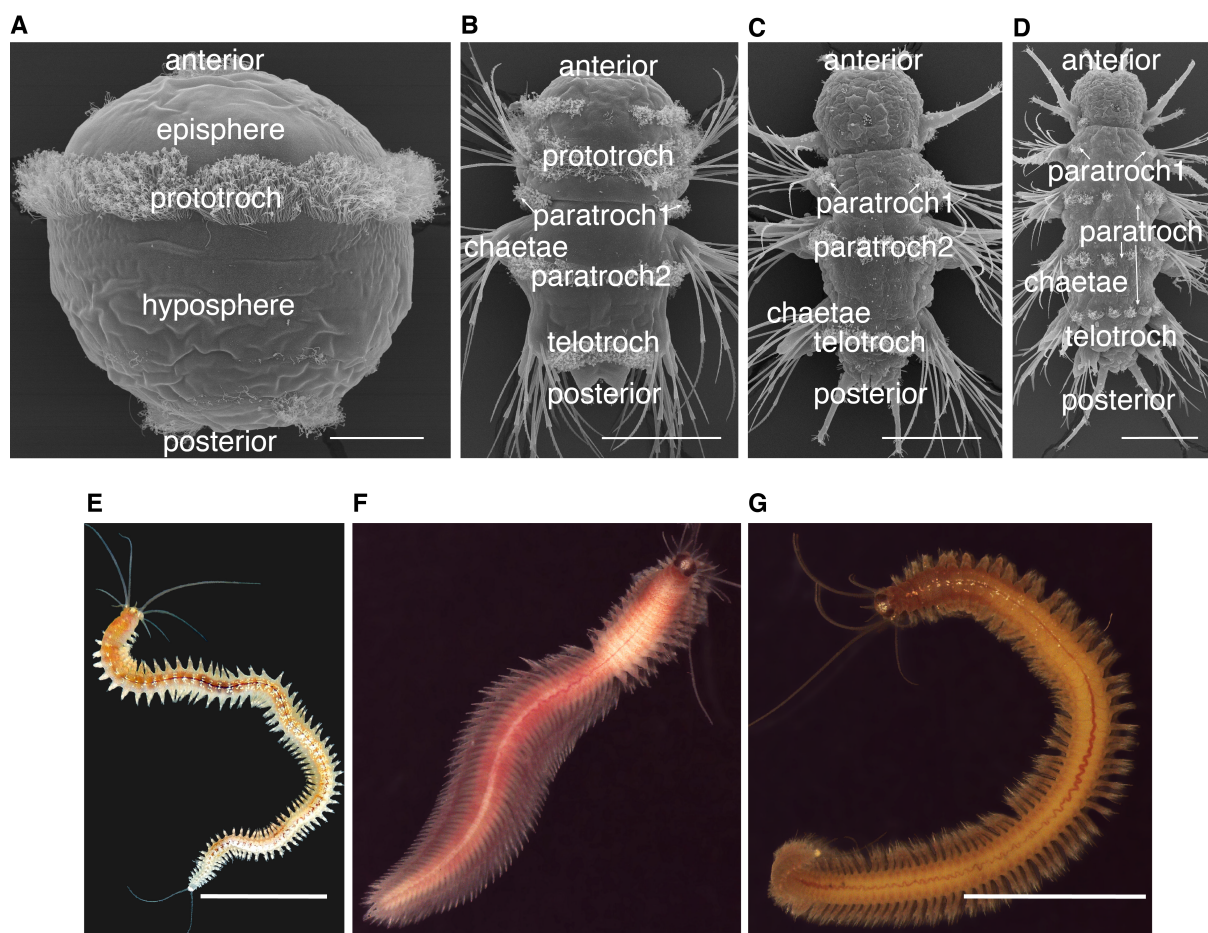


Figure 2.1.: *Platynereis* life cycle

(A) 48 hpf larva, scale bar 40 μm . (B) 72 hpf larva, scale bar 80 μm . (C) 6 days post-fertilization larva, scale bar 80 μm . (D) 4-6 week larva, scale bar 100 μm . (E) *Platynereis* atoke, scale bar 1 cm. (F) *Platynereis* mature male. (G) *Platynereis* mature female, scale bar 1 cm. (A-D) Images courtesy of Jürgen Berger. (E) Image courtesy of Elizabeth Williams. (F, G) Images courtesy of Iris Koch. hpf, hours post-fertilization.

into a planktonic trochophore with a spherical body after about 18 hours of development and

hatches supported by ciliary swimming after 24 hours post fertilization (hpf) (Figure 2.1 A). In addition to a pair of larval eyes, two pairs of adult eyes become visible at around 54 hpf. The larvae start elongating and segmenting, and ultimately settle in the benthic environment (on the sea floor) after 3-4 days of development. Segment proliferation continues throughout *Platynereis* life (Figure 2.1 B-D). After the atokous period of life (Figure 2.1 E), a large portion of which is spent in a tube, sexually mature worms leave their tubes to search for a sexual partner and spawn. Reproductive activity in *Platynereis* is highly synchronized with the moon cycle [45, 46]. Mature males are red-colored and can be distinguished from females, which are yellow-colored (Figure 2.1 F, G).

The developmental stages in *Platynereis* have been studied and differentiated based on anatomical and physiological characteristics, such as the appearance of the ciliary bands, the formation of the parapodia or the onset of feeding [47]. *Platynereis* development is highly synchronous [45], which facilitates developmental studies on this organism. The stages that I focus on in this project are:

- Late trochophore (40 hpf-48 hpf), characterized by distinct stomodeal opening and first chaetae appearing inside the trunk (Figure 2.1 A).
- Early nectochaete (66 hpf-75 hpf), with the body rapidly elongating, parapodia moving independently and the developing metatroch (Figure 2.1 B).
- Late nectochaete (5 days-7 days), where antennae elongate and larvae start feeding (Figure 2.1 C).

Monod *et al.* specialized in the molecular and cellular aspects of posterior elongation and segment addition in *Platynereis* and suggested an ancestral nature of this process [48]. Conzelmann *et al.* dedicated several studies to thoroughly investigate the role of neuropeptides in *Platynereis*, exploring the dynamics of neuropeptide expression throughout the life cycle [49] and the role of neuropeptide signaling in various life history processes, such as vertical migration [32] or settlement [33].

Platynereis was also used as an experimental organism in population genetics to understand *Platynereis* population structure [50], in toxicology to investigate the effect on *Platynereis* of the disinfection byproducts present in a sewage effluent [51], and in ecology to study acclimatization of *Platynereis* to ocean acidification [52].

Due to its highly conserved gene structure and few evolutionary changes, *Platynereis* has served as a model for several evolutionary studies [53, 54, 55, 56, 57, 58, 59, 60, 61, 62]. Raible *et al.* compared *Platynereis* and human genomes and established their ancestral nature [54]. Through an exploration of the molecular architecture of the trunk nervous system in *Platynereis*, Denes *et al.* discovered a common origin of nervous system centralization in bilateria [55].

Platynereis is well suited for assessing gene expression dynamics during micro- and macroevolutionary development [53, 54, 55, 56, 57, 58, 59, 60, 61].

2.2.2. Molecular techniques established in *Platynereis*

The *Platynereis* genome is organized into 28 chromosomes and contains 1 Gbp [63]. Large collections of sequencing data from both the laboratory strain and natural populations of *Platynereis* are available [64]. The whole genome of *Platynereis* has been sequenced and EST libraries also exist [65]. A wide range of genetic, genomic, pharmacological and other tools have been developed for *Platynereis* to allow investigation of important questions in evolution, development, gene regulation and behavior:

A robust protocol for **whole-mount *in situ* hybridization (WMISH)** in *Platynereis* [66] is widely used. *In situ* hybridization allows the localization of particular RNA or DNA sequences in a tissue. The sequence is detected by applying a complementary strand of a sequence with a reporter marker attached.

Immunohistochemistry (immunolabeling) protocol was established for *Platynereis* [67]. Immunohistochemistry localizes proteins in tissues by applying tagged antibodies specifically binding to proteins.

Gene knockdown with morpholino-oligomers was applied to *Platynereis* larvae [33]. This technique reduces expression of a gene by binding oligomers to a target messenger RNA, which prevents RNA translation.

Transposon-mediated transient and stable germline transgenesis were established in *Platynereis* [68]. Transgenesis introduces a new gene into the genome of an organism, which will be active in that organism and will be transmitted to its offspring.

Targeted mutagenesis using TALENs was described for *Platynereis* [69]. This technique is a genome editing process involving artificial restriction enzymes.

Microinjection protocol for *Platynereis* zygotes and early embryos facilitated analyzing cell lineages and cleavage patterns [33, 70, 68, 69]. Microinjection delivers a substance, such as exogenous proteins, peptides, drugs, or cDNA constructs, in a living cell using a microscopic setup.

Pharmacological experiments. Due to their permeable body wall, *Platynereis* larvae are amenable to drugs and signaling proteins added to the water. This property was exploited for analyzing the influence of chemical components and proteins on biological processes, such as larval locomotion [32] or settlement [33].

Laser-assisted [12, 71] and **chemical cell ablation** [72] have been carried out on *Platynereis*. Laser-assisted eye-ablation was applied for studying larval phototaxis.

Behavior setups The advantage of *Platynereis* is its relatively simple anatomical organization combined with the presence of sensory organs capable of detecting a wide range of environmental stimuli, such as light (phototaxis), chemicals (chemosensation), pressure (barotaxis), temperature (thermosensation) and mechanical stimuli (mechanosensation). Various behaviors were developed in *Platynereis* as a response to the stimuli, for example, vertical migration as a reaction to a pressure change [unpublished data], directional swimming to follow the gradient of a light source [12, 34] or crawling as a response to settlement cues [33].

Due to its transparent body, *Platynereis* larvae are accessible for **light microscopy**, which employs visible light and a system of lenses to magnify images of small samples. Confocal light microscopy is used to detect an *in situ* hybridization signal in *Platynereis*. Confocal microscopy uses a pinhole to increase optical resolution. A special reflection technology has been developed [73], which was recently enhanced by using TDE as a mounting medium [74]. A timelapse 3D confocal imaging was used to track the formation of *Platynereis* mesodermal cells during early development [75].

The small size of *Platynereis* and a relatively simple nervous system organization makes it attractive for **serial transmission electron microscopy (TEM)** [34]. TEM produces an image of a specimen using a beam of electrons transmitted through a specimen and provides the resolution sufficient to reconstruct neuronal architecture [12].

A slowly evolving genome, ancestral cell types, a neuropeptide complement close to vertebrates, a regeneration ability, and its circadian and lunar rhythms make *Platynereis* an attractive and useful model organism [43]. These properties have driven the development of genetic tools, such as immunohistochemistry, germline transgenesis, morpholino-mediated knockdown and microinjection. These and novel approaches and technologies will help to address new and more detailed functional biological questions using *Platynereis* as a model organism.

3. Gene expression atlas for *Platynereis dumerilii*

3.1. Introduction

Anatomical atlases represent the anatomy of a given organism placed in a coordinate system standardized for the whole population or a part of the population (e.g. of the same sex, developmental stage or experimental conditions). Atlases visualize complex anatomical relations and can provide morphological, molecular and/or functional information for the biological structures. Over the past years, traditional paper-based atlases have been enhanced via 3D digital atlases, which can be constructed based on different imaging modalities, such as magnetic resonance imaging (MRI), positron emission tomography (PET), micro-CT or light microscopy (LM). The atlases can be generated using various signals, such as stained axonal scaffold or neuropil [76, 53], nuclear stain [77, 78] or differential interference contrast (DIC) [79].

3.1.1. Anatomical atlases in solving biological questions

Anatomical atlases provide standardized anatomy that serves as a normalization procedure, therefore they are often applied to 3D parameterization of anatomical substructures [80, 81, 82, 83, 84]. Such measurements can be further used for inter-species and inter-individual comparisons or for allocating differences in brain regions and associating these differences with deviation in cognitive skills, perception, behavior or disease state. For example, Dreyer *et al.* compared standard neuropil volumes in male and female flour beetle to analyze sex-specific volumetric differences in brain regions [82]. Jundi *et al.* investigated sexual dimorphism in the sphinx moth and discovered significant differences in brain region sizes between male and female individuals [81]. Wong *et al.* performed volumetric comparisons of knockout and wild type mouse embryos to characterize genetic phenotypes [83].

The anatomical substructures delineated in the standardized atlas can also be used for automatic segmentation of individual specimens [77, 84, 85]. Muñoz-Moreno *et al.* generated probability maps for each delineated structure in the rabbit atlas and then used these maps combined with image registration to automatically segment new specimens [85]. Long *et al.* segmented and

named nuclei in *Caenorhabditis elegans* (*C. elegans*) to generate an atlas and later automatically determine nucleus identities in new specimens [77].

Atlases provide a convenient platform for characterizing temporal dynamics of the organism of interest. For example, Fowlkes *et al.* and Castro *et al.* followed the temporal dynamics of gene expression patterns in early developmental stages using *Drosophila* and zebrafish atlases respectively [86, 87].

An important benefit of anatomical atlases is the ability to map functional activation and connectivity information [88, 89]. For example, Jundi *et al.* revealed potential connection between two individual neurons involved in polarization vision in the locust [89]. Similarly, Brandt *et al.* analyzed the connectivity between olfactory neurons and neuropils using the honeybee atlas [88].

3.1.2. Image registration for atlas generation

Since atlases represent a population-based average, they are often generated based on multiple individuals. The integration of imaging data acquired across multiple individuals and experiments requires image registration methodologies. Simple superimposition of images would be insufficient due to variations in morphology and experimental conditions. Image registration methods can be classified as *intensity-based* [32, 53, 79, 85, 90] or *segmentation-based* [77, 80, 81, 86, 91]. The intensity-based registration relies on a metric calculated from pixel intensities (for example, correlation or mutual information [92]). Intensity-based image registration has been employed for generating atlases based on various reference signals, for example, the stained axonal scaffold or neuropil [[76, 53] nuclear stain [78], DIC [79], MRI [85, 90] or PET [84].

Alternatively, segmentation-based registration methods are based on prior delineation and annotation of anatomical structures [90], which then guide image registration. Cells expressing a marker gene [53], stained nuclei combined with a transgenic muscle label [77], or other anatomical structures [81] can all serve as a reference for segmentation. A special case of segmentation-based approaches is surface-based registration [93, 94, 95], which aligns the boundaries of anatomical structures using surface-fitting algorithms and then propagates transformation to the internal structures.

Image registration protocols usually start with coarse, rigid registration that is further refined by non-rigid registration. The virtual insect brain (VIB) protocol [[80]] begins with global and local rigid registration, refined by non-rigid registration. This protocol was used to generate brain atlases for the fruit fly (*Drosophila melanogaster*) [80], the sphinx moth (*Manduca sexta*) [81], the flour beetle (*Tribolium castaneum*) [82], and the desert locust (*Schistocerca gregaria*)

[96]. Another image registration protocol, the iterative shape-averaging method (ISA) [88], EMPLOYS affine registration followed by iterative non-rigid registration. This protocol was used to generate atlases for the honeybee (*Apis mellifera*) [88], the desert locust (*S. gregaria*) [96], and the tobacco budworm (*Heliothis virescens*) [91].

3.1.3. Extended atlases

Atlases show their full potential when extended with additional information such as morphology of individual neurons or gene expression patterns. Such atlases overcome the limitations of traditional single- or multicolor labeling light microscopy methods, which are often restricted in the number of channels (gene expression patterns, cells) that can be simultaneously detected (an exception is Brainbow technology [97]).

Neuronal connectivity atlases provide insight into the organization of neural circuits and how these circuits regulate animal behavior. Such atlases are thus needed to formulate hypotheses about the information flow within neuronal networks and to guide experimental manipulations aimed at understanding how neuronal circuits modulate animal behavior. Neuronal connectivity atlases have been generated for the honeybee (*Apis mellifera*) [88], the desert locust (*S. gregaria*) [89, 96] and the fruit fly *Drosophila melanogaster* [76].

In turn, gene expression atlases provide insight into the relationship between molecules and behavior. Gene expression atlases play an important role in correlating gene expression with more salient characteristics such as morphology, neuronal connectivity or physiology. They elucidate gene regulatory interactions [86], inter-individual variation in gene expression [80] or the molecular fingerprint of cell types [77], and allow evolutionary comparisons [53]. Similar to neuronal connectivity atlases, they provide a basis for functional manipulations of an organism under study. Gene expression atlases have been generated for *Drosophila* [86], mouse [79], chicken [98], zebrafish [78] and *Platynereis* [53].

Mapping additional features to the reference template requires labeling of specimens with a reference marker and a marker of interest. Markers of interest can include neuron-filling dyes labeling individual neurons [88, 91, 96] or fluorescent proteins expressed in a subset of cells by transgenic techniques [76]. Gene expression patterns can be visualized by *in situ* hybridization [79, 53, 86], immunostaining [86], or transgenic techniques [80].

3.1.4. Current work contribution

Here, I present a whole-body gene expression atlas for the early developmental stages of *Platynereis*. *Platynereis* exhibits stereotypic early development, which together with a small size constitute important prerequisites for atlas generation. Despite this, image registration in

Platynereis has so far been restricted to the 48 hours post fertilization (hpf) trochophore larval episphere [53], due to the limited depth-penetration achieved by using imaging setups with refractive index mismatches (glycerol mounting with an oil immersion objective). The previous image registration procedure starts with rigid registration, followed by affine and B-Spline deformable registration as implemented in the Insight Toolkit (ITK). It employs an iterative approach for generating a template for the larval episphere using an acetylated-tubulin signal as a reference marker for registration.

In the present study, I extend the existing image registration protocol and apply it to the *Platynereis* full-body images from whole-mount *in situ* hybridization and confocal microscopy in order to generate a gene expression atlas for the early developmental stages (48 hpf, 72 hpf and 6 days post fertilization, dpf) of *Platynereis*.

The gene expression atlas assists in solving an important problem in neuroscience: associating anatomical regions with the respective functionality. To facilitate this task, I implement software tools to extract behavioral and functional information associated with a gene and subsequently to annotate the corresponding gene expression patterns in the atlas with the extracted functional information.

To provide a convenient atlas representation for exploring gene expression patterns, I modeled a gene expression atlas in Blender, 3D graphics software, which encompasses various options to store, visualize, manipulate and analyze gene expression patterns. I enable various querying and visualization options, which facilitates detecting coexpressing and therefore possibly interacting molecules.

3.2. Methods

3.2.1. *In situ* hybridization

The published *in situ* hybridization protocol was performed using nitroblue tetrazolium (NBT)/5-bromo-4-chloro-3-indolyl phosphate (BCIP) staining combined with anti-acetylated tubulin and DAPI staining as described in ref. [73] with the following modification: larvae after *in situ* hybridization were transferred into 97% TDE diluted with PBS plus 0.2% Tween. *In situ* hybridization preparations were performed by Markus Conzelmann, Elizabeth Williams, Csaba Veraszto, Nadine Randel, Luis Bezares and other members of the Jékely group.

3.2.2. Immunohistochemistry

Immunostaining was performed as described in the previously published protocol [99] using affinity-purified rabbit neuropeptide antibodies combined with mouse anti-acetylated tubulin antibody and DAPI staining. Immunostaining was performed by Markus Conzelmann.

3.2.3. Confocal microscopy

The prepared larvae were placed between a glass slide and a coverslip for imaging. The larvae were imaged on an Olympus FV1000 confocal system on an IX81 inverted microscope using 0.15 mm working distance and an UPlanSApo 60x Oil objective with 1.35 N.A. (Olympus Deutschland GmbH, Hamburg, Germany). Diode lasers 405, 559 and 635 nm were used for simultaneous three-channel excitation. Channel 1 for DAPI detection had a 425 to 475-band pass filter, channel 2 for TRITC detection had a 570 to 625 band pass filter and channel 3 for the detection of NBT/BCIP had a 780 long-pass filter. Images were scanned at 1.1x zoom and a pixel size of $0.414 \mu\text{m} \times 0.414 \mu\text{m}$, resulting in stack of images of size 512×512 pixels (Figure 3.1). The entire volume of the larvae was scanned with a z-step size of $0.41 \mu\text{m}$, to obtain an isotropic voxel size.

Confocal microscopy was enhanced closer to its depth-limit by using 2,2'-thiodiethanol (TDE) as a mounting medium, the refractive index of which matches to that of glass and immersion oil [83]. The use of TDE enabled high-quality whole-body scans of *Platynereis* larvae at the early developmental stages (48 hpf up to 6 dpf) using isotropic voxel size. Confocal microscopy was performed by myself, Aurora Panzera, Elizabeth Williams, Csaba Veraszto, Nadine Randel, Luis Bezares and other members of the Jékely group.

3.2.4. Image registration

Image registration was applied to the image stacks obtained from *in situ* hybridization samples imaged by confocal microscopy. The implemented image registration protocol subsequently applies rigid, affine and deformable transformations using an intensity-based metric. The protocol was implemented using ImageJ / Fiji [100, 101] for rigid registration and the Insight Segmentation and Registration Toolkit (ITK) [102] for non-rigid registration.

Rigid registration Two approaches were developed to rigidly align image stacks. One employs the Rigid Registration plugin [103] available in Fiji, which implements iterative optimization to register images to a template. The other is a custom plugin that orients images based on anatomical features obtained from the acetylated-tubulin signal and DAPI stained nuclei. The antero-posterior orientation for 48 hpf larvae was defined based on the prominent signal of the prototroch ciliary band. The position of the ventral nerve cord determined the orientation. The antero-posterior orientation for 72 hpf larvae, which have elongated body, was defined based on the major body-axis. The dorso-ventral orientation was identified by the position of the nerve cord. The plugin was implemented using ImageJ functionality.

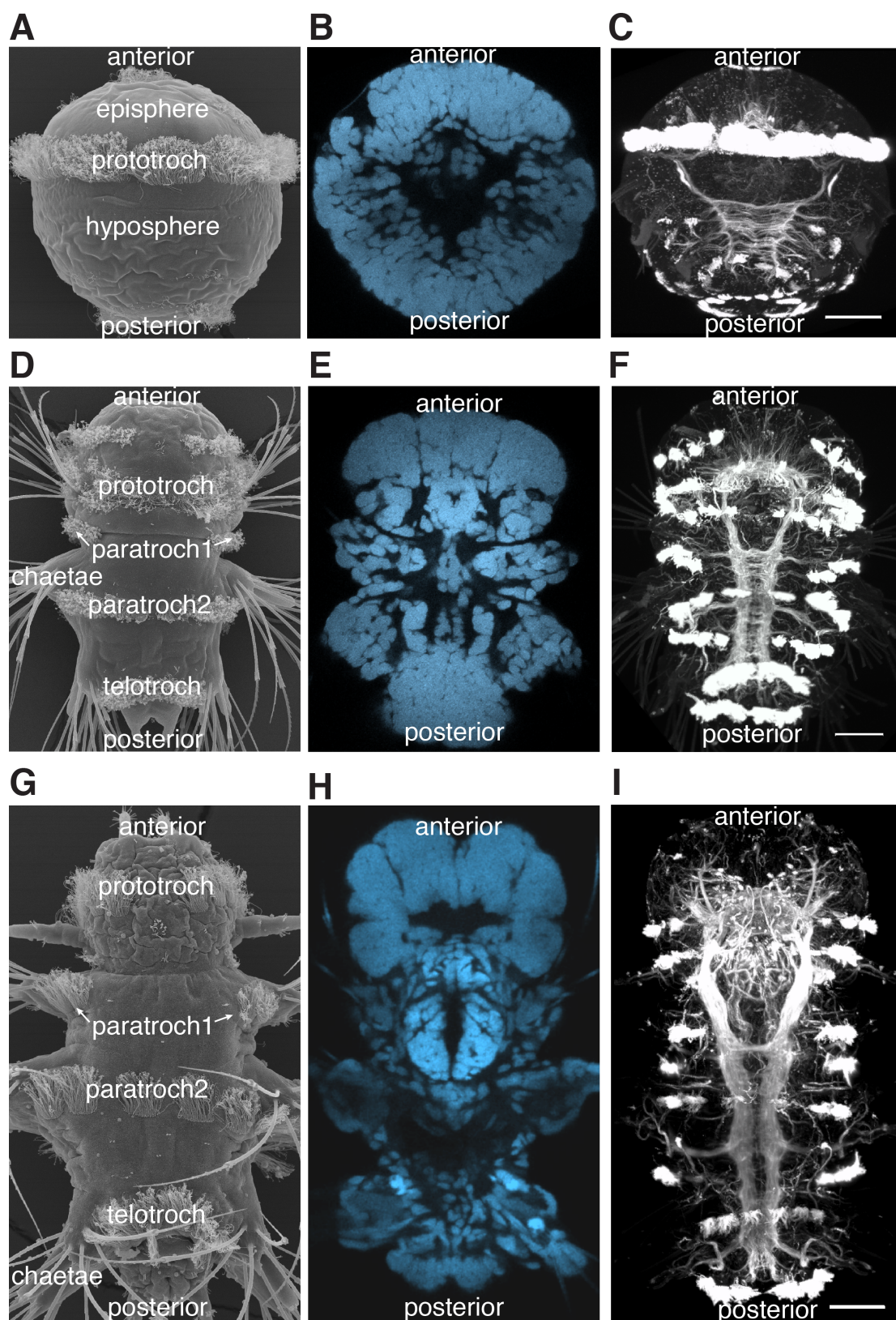


Figure 3.1.: Scanning electron microscopy and light microscopy images of the *Platynereis* larvae.

(A, D, G) Scanning electron microscopy images of 48 hpf (A), 72 hpf (D) and 6 dpf larvae (G). Scanning electron microscopy images courtesy of Jürgen Berger. (B, E, H) Single sections from the image stacks of DAPI-stained specimens at 48 hpf (B), 72 hpf (E) and 6 dpf (H). (C, F, I) 3D views of tubulin-stained specimens at 48 hpf (C), 72 hpf (F) and 6 dpf (I). Scale bars 30 μ m. hpf, hours post fertilization. dpf, days post fertilization.

Non-rigid registration The initial rigid registration was refined using affine and deformable registration implemented in the ITK library. The following parameters of the ITK image registration were specified:

- the interpolator that estimates pixel intensities at non-grid positions after transformation (ITK class `LinearInterpolateImageFunction`)
- the optimizer that searches for the optimal transformation for registration (ITK class `RegularStepGradientDescentOptimizer`)
- the metric that evaluates the alignment at each optimizer step (ITK class `MattesMutualInformationImageToImageMetric`).

The Mattes mutual information metric (MMI) [95], which extends the mutual information metric, was employed. MMI is evaluated based on a subset of pixels uniformly sampled from the image. The main difference to the `MutualInformationImageToImageMetric` class in ITK is that the subset of pixels sampled at the first optimizer iteration is reused in the subsequent iterations when calculating the MMI metric.

B-Spline (ITK class `BSplineDeformableTransform`) and affine transforms (ITK class `AffineTransform`) were used for the affine and deformable registration steps respectively. The multi-resolution approach (`MultiResolutionImageRegistrationMethod` class) was utilized, which registers images first at a low resolution and then stepwise at higher resolutions. Such approach helps the optimizer to avoid local minima. Non-rigid registration of the whole-body *Platynereis* images takes approximately 50 min on a PC with 32Gb RAM and an i5-2500 CPU 3.3GHz x 4 processor.

3.2.5. Template generation

I implemented an iterative procedure for template generation similar to the procedures employed in previously published methods [53, 88] (Figure 3.2). Template generation required a set of specimens stained with a reference marker (50 specimens were employed). A special procedure was applied to avoid bias towards the first selected image. First, a median-size specimen was selected. The median image was then registered to the rest of the original images and the average of the obtained transforms was calculated. The average transform was applied to the median image, which was then subjected to the iterative procedure for template generation. In the first iteration, the original images were registered to the transformed median image and sorted according to the similarity to this image (Mattes mutual information metric). The top 40 scans were averaged to generate a template for the following iteration. Three iterations were performed.

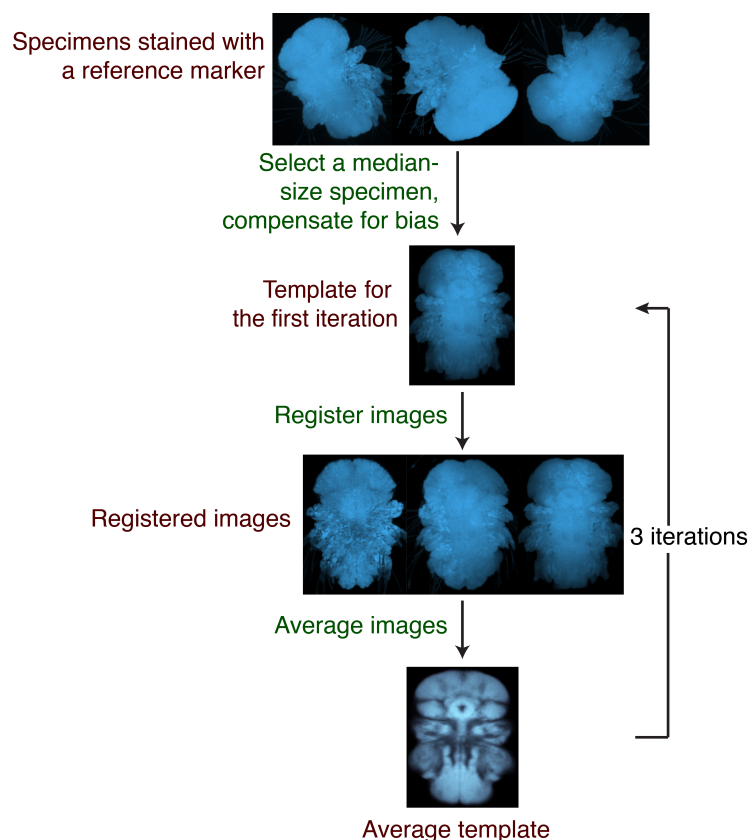


Figure 3.2.: A pipeline for template generation.

A median-size specimen was selected from the specimens stained with the reference marker (acetylated tubulin or, as here, DAPI nuclear stain). The bias-compensation procedure was then applied to the median-size specimen. Next, the original images were registered to the transformed median image and averaged. The average image generated in each iteration was used as a template in the next iteration. The procedure was repeated three times.

3.2.6. Gene expression pattern registration

The generated template served as a reference space to register gene expression patterns. All specimens were stained with a reference marker and a marker of a gene of interest. The reference marker was registered to the template and the obtained transform was applied to the corresponding gene expression pattern (Figure 3.3). To avoid bias towards the gene expression pattern of an individual specimen, multiple specimens were stained with the same gene marker and subsequently registered and averaged.

3.2.7. Multichannel visualization

I developed an ImageJ plugin, ChannelMerger, to simultaneously visualize multiple registered gene expression patterns. The plugin merges individual channels into one RGB stack and provides adjustment options such as color change and visibility setting. The merged RGB image is compatible with ImageJ functionality such as saving as tif stack, adjusting brightness/contrast,

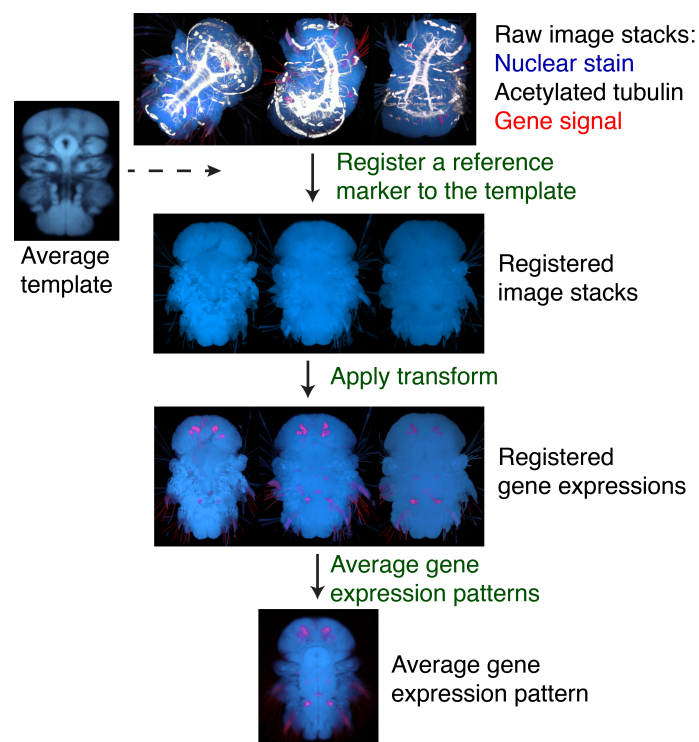


Figure 3.3.: Gene expression pattern registration pipeline

The pipeline accepts as an input individual specimens stained with a reference marker (acetylated tubulin or DAPI nuclear stain) and a marker of a gene of interest. The reference marker is then registered to the template and the obtained transform is applied to the corresponding gene signal. Several specimens are stained with the same gene marker, registered to the template and averaged to obtain a representative gene expression pattern.

viewing in 3DViewer, and others. The ChannelMerger plugin is available for download [104].

3.2.8. Model for gene expression atlas

The generated gene expression atlas was modeled using Blender 3D graphics software. Each registered gene expression pattern was converted to its surface representation using ImageJ 3D Viewer and subsequently imported in Blender. Broad gene expression domains were then smoothed using Gaussian filter with sigma in the range of [1,2]. Cell-specific domains were approximated with spheres (Figure 3.4). An averaged signal of the registered tubulin marker was converted to surface representation and imported into Blender to provide an anatomical reference for the gene expression atlas.

The atlas was extended with querying by gene name, gene function and gene type using the embedded Python scripting. The imported gene expression domains were annotated by experts with gene name, associated functionality, and gene type. Such annotations were implemented using the Blender grouping option. Expression domains associated with the same gene, the same gene type or functionality were grouped together. The names of the gene-function groups

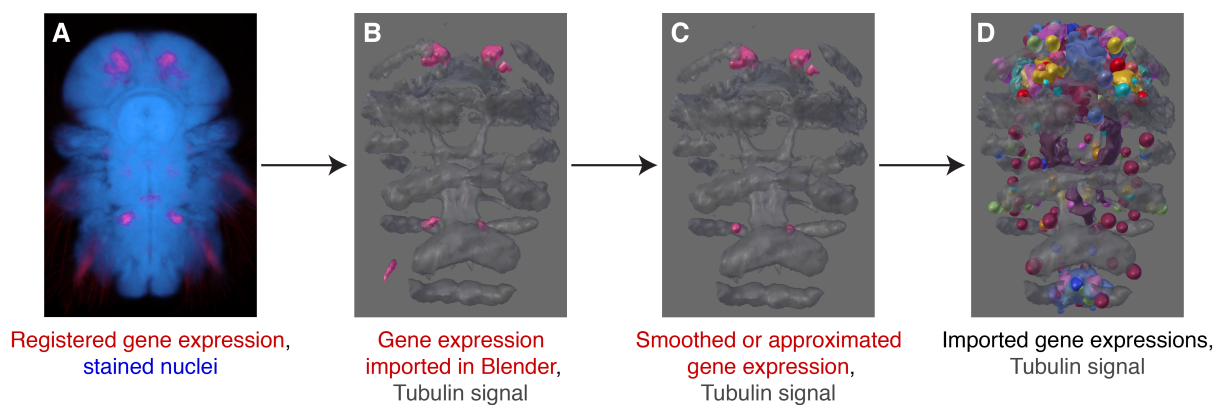


Figure 3.4.: Exporting gene expression patterns in Blender.

An intensity-based representation of an average gene expression pattern (A) is converted into a surface representation (B) using 3D Viewer. The surfaces are then exported in OBJ format and imported into Blender (C), where the broad gene expression domains are smoothed and the cell-specific domains are approximated with spheres (D).

start with an underscore “_”, the names of the gene-type groups start with a hyphen “-” and the gene names can start with any other character. Blender native classes can be extended with custom properties of any type. Custom property is not stored in the project file and needs to be redefined after restarting. Blender class *Group* was extended with a custom boolean property *Visibility*, which was used to implement showing and hiding groups of objects according to queries over different groups (Figure 3.5).

The analysis of gene expression colocalization was implemented using the *Boolean* modifier in Blender, which creates a single compound out of two objects using difference, union or intersection operations. The intersection operation was used for gene expression colocalization analysis. All the expression domains of the two specified genes are examined for colocalization using the *Boolean* modifier and the intersection volumes are displayed.

3.2.9. Mobile GeneExpressionAtlas

The modeled gene expression atlas was implemented as an application for Android [105] mobile devices named Mobile GeneExpressionAtlas. It was developed using an ADT Bundle, including Android Software Development Kit and a version of Eclipse Integrated Development Environment [106]. The libGDX game development framework [107] was used to visualize the 3D model in the application. The Blender model was optimized for mobile devices by reducing the number of the vertices in the model using the *Decimate* modifier in Blender. The model was exported in OBJ format, converted into GD3D format using the FBX converter [108] and then loaded in the mobile application using libGDX.

The developed application begins with the *MainActivity* class, which extends *AndroidAppli-*

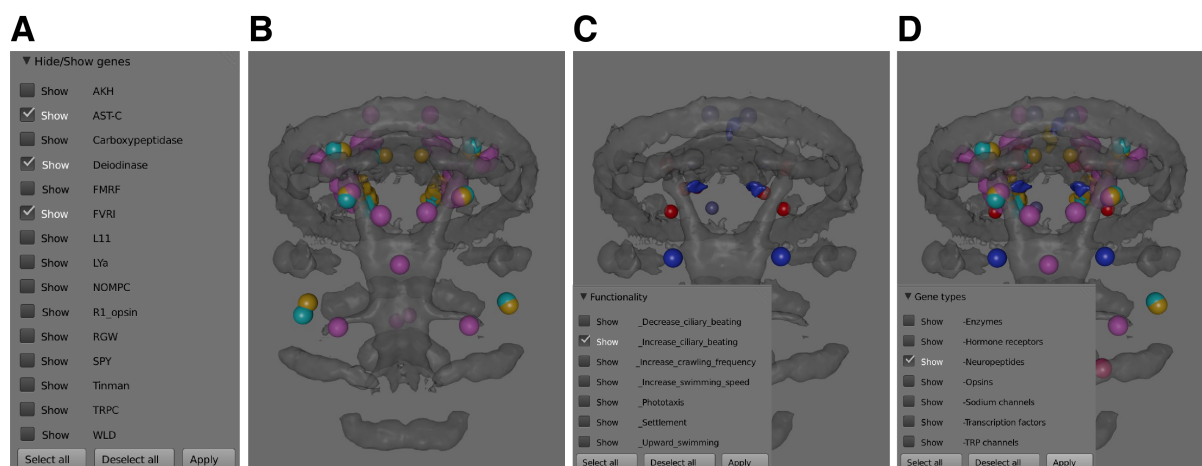


Figure 3.5.: The interface implemented for the model.

(A, B) Querying gene expression patterns by name. The menu (A) lists all available gene expression patterns that can be shown or hidden in the 3D view. The 3D view (B) of the model shows the gene expression patterns selected in (A). (C) Querying gene expression patterns by molecule functionality. The 3D view shows the expressions of the genes associated with "increasing ciliary beating", that has been selected in the menu. (D) Querying gene expression patterns by type. The 3D view shows expression domains of the type 'neuropeptide', as indicated in the menu.

cation class from the libGDX library (Figure 3.6). The *ModelVisualization* class, which extends libGDX *Game* class, controls the application logic and switching between the main menu (*MainMenuScreen* class) and the 3D view (*ModelScreen* class). The main menu provides a list of checkboxes to specify the show/hide option for molecules of interest (Figure 3.17). The model is asynchronously loaded using the libGDX class *AssetManager* and then displayed in a new screen implemented in the *ModelScreen* class (Figure 3.6). The *Renderer* class implements functionality for rendering and manipulating the model. The visibility setting is implemented using material *BlendingAttribute*, where the field *alpha* of this attribute is set to one for the groups of objects that are checked as visible in the main menu and zero for the objects set as hidden. Minimal Android version is 2.0 (API level 5). The application ran successfully on the Samsung Galaxy Note 10.1 tablet and Samsung Galaxy S3 Mini.

3.2.10. Online GeneExpressionAtlas

The modeled gene expression atlas was implemented as a web application named Online GeneExpressionAtlas. The web application was developed using the HTML and JavaScript [109] programming languages and the Three.js library [110]. First, the Blender model was optimized for a web application by applying the *Decimate* modifier to each object, which reduced the number of the vertices in the model. It was then exported from Blender in .js (javascript) for-

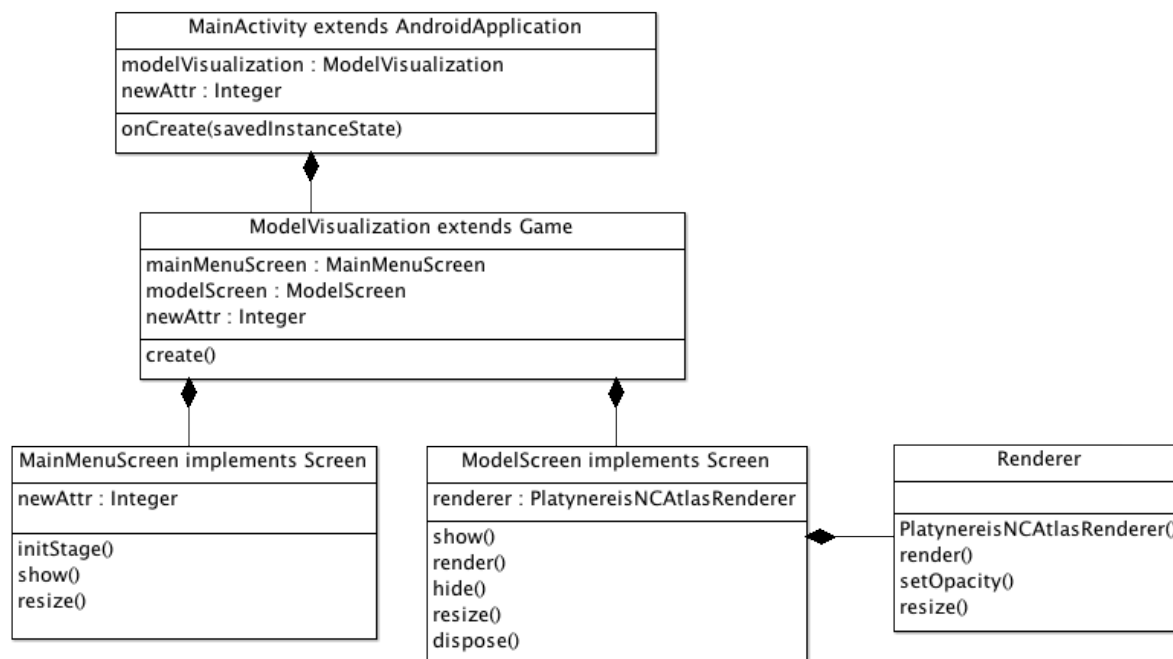


Figure 3.6.: Class diagram for the Mobile BrainDiscoverer application.

The *MainActivity* class is initialized when the application starts. It contains an instance of the *ModelVisualization* class, which is responsible for managing the application logic and switching between the main menu (*MainMenuScreen* class) and the 3D view (*ModelScreen* class). The *Renderer* class implements functions for rendering and manipulating the 3D view objects.

mat using the Blender add-on Three.js [111]. The model was then loaded in the application using the Three.js library that enabled 3D visualization. The application was extended with an option to manipulate the model view using the OrbitControls library in Three.js. The option to set the visibility setting for the individual gene expression patterns was implemented. The visibility setting can be set using the list of checkboxes on the right (Figure 3.18 B). Online GeneExpressionAtlas requires WebGL library [112] enabled in a web browser.

3.2.11. Pharmacological experiments

Platynereis larvae at 6dpf were obtained from a breeding culture (Tübingen line). Behavioral experiments were performed in seawater with 5 to 50 μ L neuropeptide added. Larvae were incubated for 2-5 min in a solution and then placed between a microscopy slide and a coverslip for recordings (Figure 3.7 A). Behavioral experiments were performed by Jan Schlüsener.

A custom ImageJ plugin was developed to measure neuropeptide effect on locomotion. The plugin classifies and characterizes the locomotor activity of *Platynereis* larvae from the video recordings. To generate a training set, body-shape descriptors were extracted for each larva

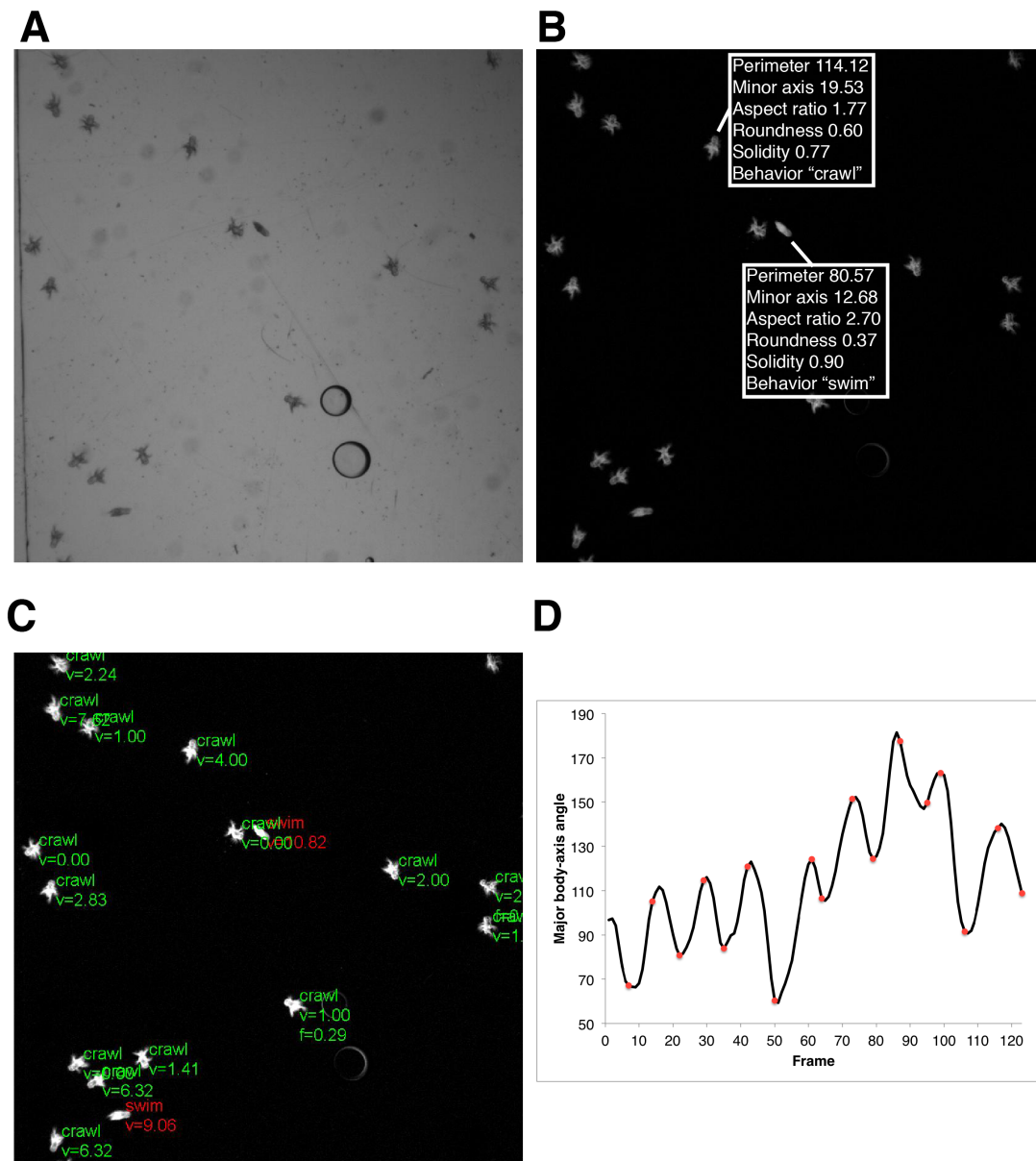


Figure 3.7.: *Platynereis* locomotion classification and characterization.

(A) A frame from a video recording of the behavioral experiment. (B) The segmented larvae from the video frame in (A). The annotation with the corresponding shape descriptors and the locomotor activity are shown for two larvae. (C) The segmented larvae from the video frame in (A) automatically annotated with their locomotor activity and locomotion characteristics. 'swim' or 'crawl', current locomotor activity; 'v', swimming or crawling speed; 'f', crawling frequency. (D) Body angle fluctuations for a single larva (black line) and the larva bending events detected by an expert from the video recording (red dots).

using an ImageJ plugin ParticleAnalyzer and the executed locomotor activity was visually determined for each larva. The relevant descriptors were selected using Best First method in Weka, resulting in a set of the following descriptors:

- Perimeter of the body
- Secondary axis of the fitted ellipse
- Aspect ratio of the fitted ellipse, calculated as: $AR = \frac{[Primary\ axis]}{[Secondary\ axis]}$
- Roundness of the body area: $R = 4 * \frac{[Area]}{\pi * [Primary\ axis]^2}$
- Solidity of the body area: $S = \frac{[Area]}{[Convex\ area]}$

The plugin employed the Weka [113] implementation of the Naive Bayes classifier to differentiate between swimming and crawling larva activities. The Naive Bayes model was trained on the annotated data (Figure 3.7 B) and then used to characterize larval locomotion.

The locomotor activity was further characterized. Crawling and swimming speeds were calculated as the average displacement of larvae per frame. Crawling frequency was defined based on the fluctuation of the angle between the major body-axis and the X-axis. The minimum and maximum values in the body-angle fluctuation correspond to a body-bending event (Figure 3.7 D), based on which crawling frequency was calculated (Figure 3.7 C). To obtain more reliable results, only those larvae that performed more than 10 bends contributed to quantifications.

3.3. Results

3.3.1. Evaluation of image registration

To integrate multiple gene expression patterns in a common reference space, I employed image registration and generated gene expression atlases for three developmental stages of *Platynereis dumerilii*. The atlases were generated based on the image stacks obtained from the *in situ* hybridization samples imaged by confocal microscopy. I extended the existing image registration protocol, which subsequently applies rigid and non-rigid (affine and deformable) registrations. I implemented two approaches for the rigid registration step: the feature-based and the intensity-based methods (cf. Image registration). The feature-based method employs anatomical landmarks to find dorsal-ventral and anterior-posterior orientations. The intensity-based approach is based on the Rigid Registration plugin implemented in Fiji. I evaluated both methods by running them on sets of images of *Platynereis* larvae at three developmental stages: 48 hours post fertilization (hpf), 72 hpf and 6 days post fertilization (dpf). The landmark-based method was applied to the 48 and 72 hpf larvae and the intensity-based method was applied to all three

Stage	Method	Correctly oriented, %	Images Tested	Time per image, sec
48 hpf	intensity-based	9	22	4.5
	Landmark-based	86	22	1.2
72 hpf	intensity-based	90	29	12.5
	Landmark-based	76	32	1.7
6 dpf	intensity-based	88	26	10.8
	Landmark-based	-	-	-

Table 3.1.: Evaluating the intensity- and landmark-based methods for rigid registration.

stages. Since the intensity-based method required a template for registration, I selected an individual image for each stage and manually oriented it. The landmark-based method was more time-efficient for all stages and performed significantly better than the intensity-based method on the 48 hpf larvae, whereas the intensity-based method performed better on the 72 hpf larvae (Table 3.1). Therefore, I further used the landmark-based method for 48 hpf and the intensity-based method for 72 hpf and 6 dpf stages.

I also evaluated the non-rigid registration using Mattes Mutual Information metric (MMI) as implemented in the ITK toolkit. MMI is a negative-value metric and decreases with increasing image similarity. Since the metric is not comparable across image sets of different developmental stages, I explored the metric behavior in the context of the different developmental stages. I selected a median-size DAPI-stained specimen for each stage and covered a certain percentage of the volume of the selected image with random pixel values (Figure 3.8 A). The coverage percentage varied from 10 to 70%. I calculated the metric between each modified image and the original image (Figure 3.8 B). The metric decreased with the increasing volume of the random pixels, which reflected the lower similarity of the compared images.

After obtaining metric values for each developmental stage, I could use the information on the metric behavior to evaluate deformable registration. I selected 50 DAPI-stained specimens for each developmental stage and registered them to the median-size specimen of the corresponding developmental stage. I calculated the MMI metric between each sample image and the median image (Figure 3.8 D, E, F, Table 3.1). The average metric value for 48 hpf stage is -0.63, which lies closely to the 30% mark in the plot of random coverage metric (Figure 3.8 D). This can be interpreted as follows: about 70% of the sample image volume matches perfectly to the median image (70% registration match). The metric values for 72 hpf and 6 dpf are -0.54 and -0.48 respectively, which corresponds to 60-70% registration match (Figure 3.8 E, F). Setting requirement for the registration match to 60%, would result in the metric thresholds of -0.483 for 48 hpf, -0.482 for 72 hpf and -0.448 for 6 dpf larvae images. Regarding these thresholds, all specimens selected for the 48 and 72 hpf stages and 90% of the specimens at 6 dpf were registered with a registration match of 60% or higher.

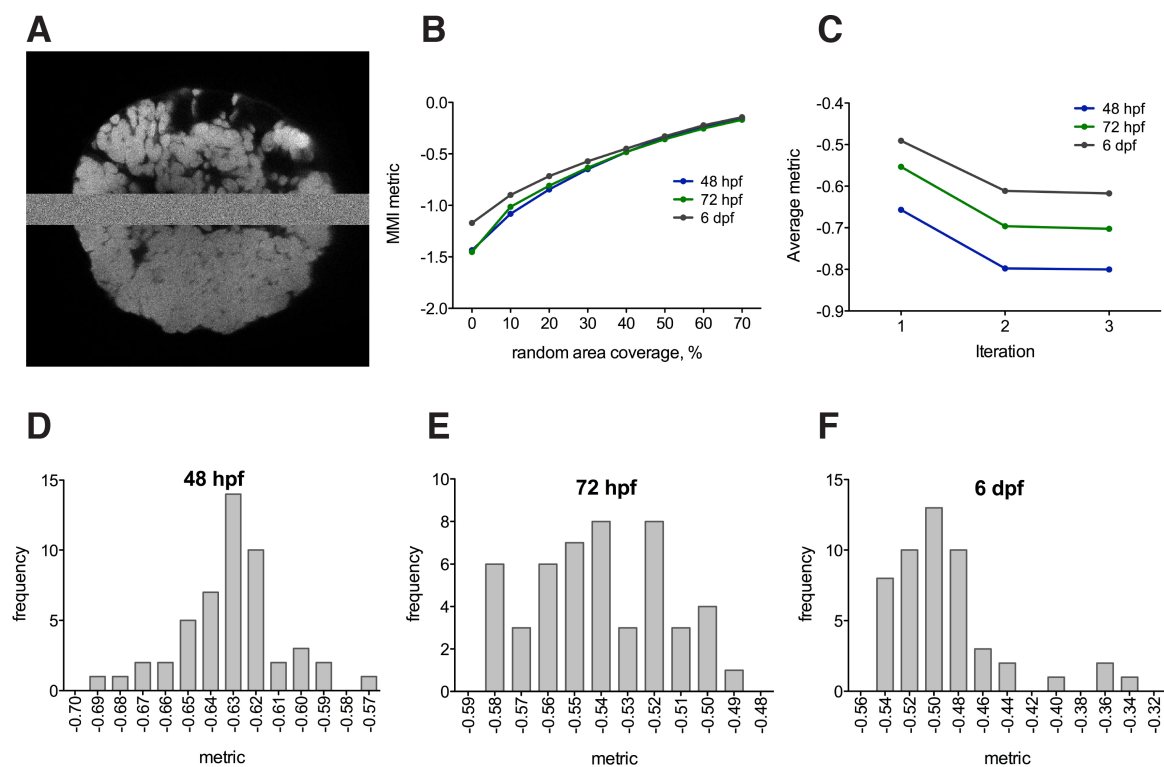


Figure 3.8.: Evaluation of non-rigid image registration.

(A) A layer from the image stack of a DAPI-stained specimen as 48 hpf. 10% of the image volume is filled with random pixel values. (B) Similarity metric behavior on the images of the DAPI-stained specimens at three developmental stages. (C) Average image-registration metric after each iteration of the template generation. (D, E, F) Distribution of the similarity metric values for 50 specimens selected for each stage. hpf, hours post fertilization. dpf, days post fertilization.

I visually inspected the registration results for the images registered with the metric above the threshold and found that the corresponding anatomical parts in the template and the sample images largely overlap after registration without introducing large distortions in the sample's anatomical structure (Figure 3.9).

3.3.2. Average anatomy templates for *Platynereis dumerilii*

Bias-compensation procedure A popular approach to generate an average anatomy template for gene expression registration is an iterative method. This method was proposed by [88] and was employed in [53, 74]. The approach suggests that a single image should be selected from a set of original images as an initial reference template. Such procedure, however, leads to bias towards the initial selected specimen [81]. Here, I introduced a procedure to minimize the bias towards the first selected specimen. First, a median-size specimen was selected and registered to the original images. The transforms obtained after registration were averaged and applied to the median image, in order to compensate for possible uncommon features of the selected

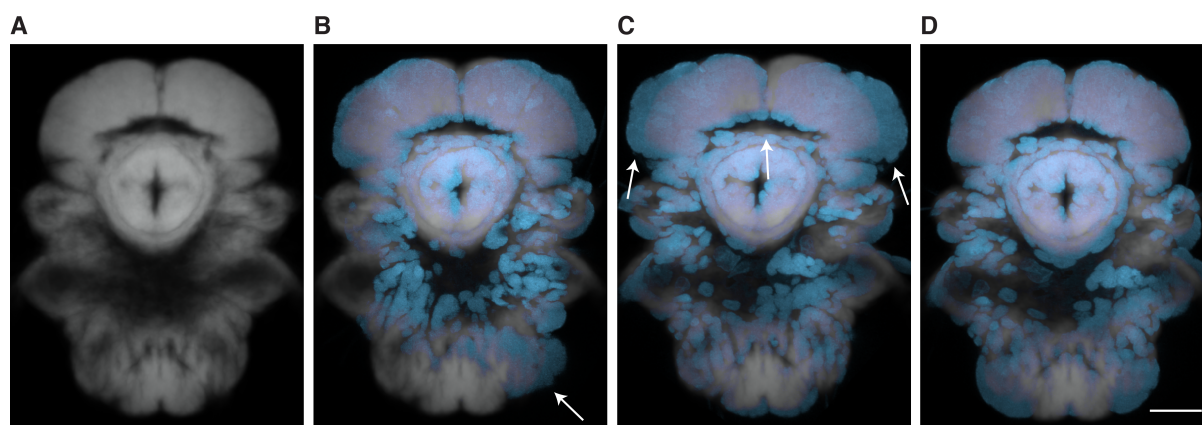


Figure 3.9.: Visual inspection of the registration results.

(A) A layer of the DAPI-template image stack at 72 hpf. (B) Overlaid layers of the DAPI-template image and the sample image after rigid registration. (C) Overlaid layers of the DAPI-template image and the sample image after affine registration. (D) Overlaid layers of the DAPI-template image and the sample image after deformable registration. Arrows indicate mismatch regions. hpf, hours post fertilization.

specimen. The transformed image was then used as a template for the first iteration. Each subsequent iteration used the average template generated in the preceding iteration as a reference template (Figure 3.2).

I also analyzed how many iterations are sufficient for generating a reliable template. I calculated the MMI metric between each sample image and the corresponding reference template and plotted the average value after each iteration for each developmental stage (Figure 3.8 C). The average metric reduces significantly ($p < 0.0001$) after the second iteration and non-significantly after the third iteration in each developmental stage. Therefore, I performed 3 iterations for generating templates.

Selection of a reference marker I selected a best-performing reference marker for generating the gene expression atlases in *Platynereis*. Markers of choice were DAPI stain, labeling cell nuclei (Figure 3.1 B, E, H), and an antibody against acetylated-tubulin, labeling neurites and cilia (Figure 3.1 C, D, F).

I generated tubulin and DAPI templates for 48 and 72 hpf larvae as described in the Methods (Figure 3.10). To assess the templates, I registered individual specimens to the templates using the respective reference markers and calculated absolute deviation of the corresponding cell positions in the individual specimens after registration. Absolute deviation was calculated as the mean of the distances of the individual cell-center positions to their average coordinate. I selected 14 cells for each 48 and 72 hpf developmental stages and 10 cells for 6 dpf. The cells were labeled with cell-specific gene markers and located in different parts of the episphere and

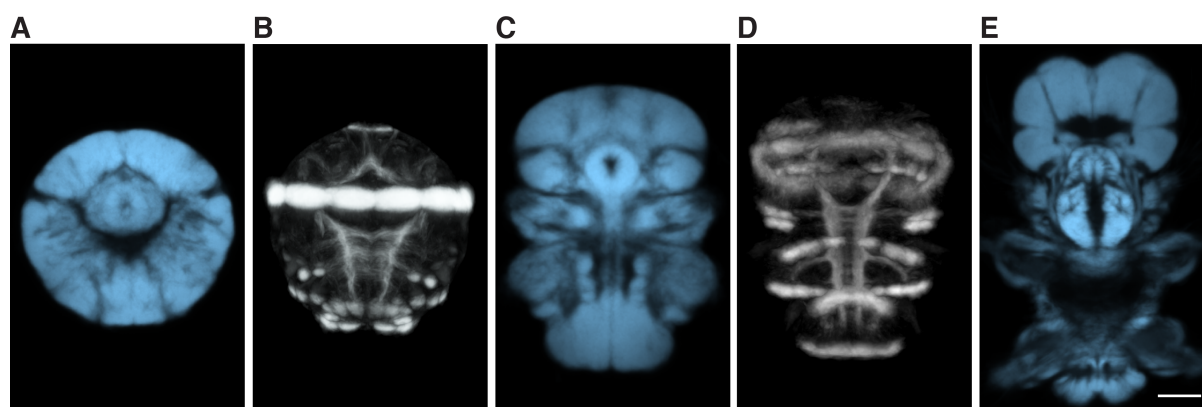


Figure 3.10.: Generated templates based on DAPI staining and acetylated tubulin. (A, C, E) average templates based on DAPI staining for 48 hpf (A), 72 hpf (B) and 6 dpf (E). (B, D) Average templates based on acetylated tubulin for 48 hpf (B) and 72 hpf (D).

the trunk (Figure 3.11 A-F). 10 specimens were stained and scanned for each gene expression in the 48 and 72 hpf stages. The results (Figure 3.11 G, H) revealed that the absolute deviation of cell positions was lower than the cell diameter (on average $10 \mu\text{m}$) for both DAPI and tubulin reference markers. Similar values were obtained for the 6 dpf DAPI template (Figure 3.11 I). However, the DAPI templates performed equally well or better than the tubulin templates, hence DAPI staining was further used as a reference marker for generating gene expression atlases.

3.3.3. Gene expression pattern registration

Next, the generated 48 and 72 hpf DAPI templates were used to register gene expression patterns as described in the Methods. To obtain a representative gene expression pattern, several specimens stained with the same gene marker were used. To find out how many original scans are sufficient to accurately register a single gene expression pattern, I selected a set of registered specimens stained with the same gene marker, then sampled subsets of different sizes from this set and analyzed how the average gene expression pattern of the subset changed with the sample size.

For the analysis, I selected two cells with the highest and two cells with the lowest absolute deviation shown in the Figure 3.11 for the 48 and 72 hpf stages. For each cell, I sampled random subsets of different sizes (one to nine, 1000 iterations for each size) from the specimens in which this cell was labelled. I calculated the average cell-body coordinate for each subset and defined the distance between the subset average coordinate to the average coordinate of the complete set. An average distance over 1000 iterations for every subset size was plotted for all selected cells (Figure 3.12). The average distance reduced with the sample size and reaches

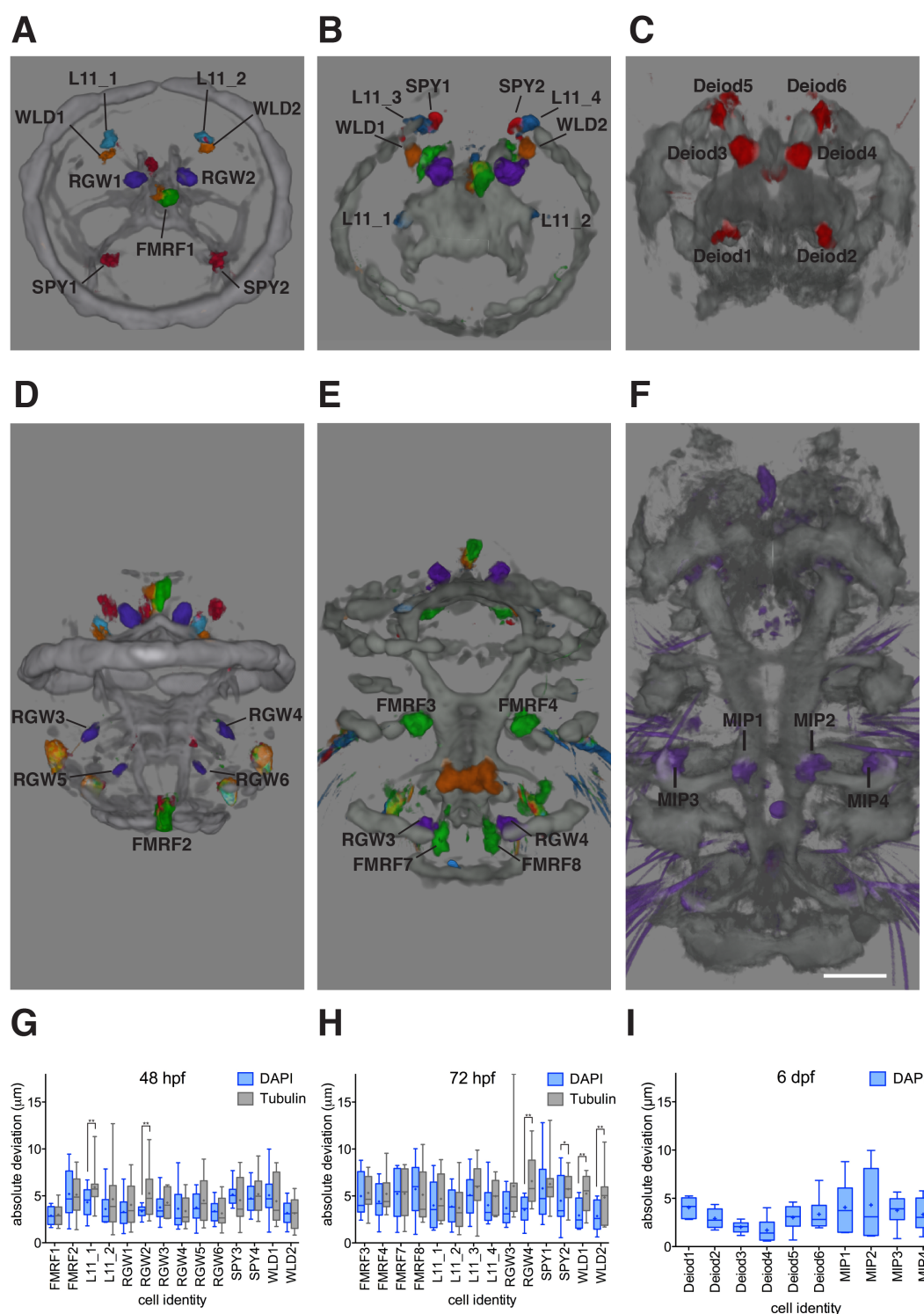


Figure 3.11.: Evaluation of the accuracy of the gene expression pattern registration to the DAPI and tubulin templates.

(A-F) Gene expression patterns and acetylated tubulin registered to the DAPI-templates for 48 hpf (A, D), 72 hpf (B, E) and 6 dpf larvae (C, F). (A-C) are anterior views, (D-F) are ventral views. The cells used for quantifications are indicated. (G, H, I) Absolute deviation of the corresponding cells (that is, distances of the individual cells to their average coordinate) for 48 hpf (G), 72 hpf (H) and 6 dpf (I) larvae for the indicated in (A-F) cells. The DAPI templates are evaluated at all three stages and the tubulin templates are evaluated at 48 and 72 hpf. The graphs represent min-max values, (+) indicates mean values. p-values of a paired t-test are shown: * $p < 0.05$, ** $p < 0.01$. MIP, myoinhibitory peptide; hpf, hours post fertilization; dpf, days post fertilization. Scale bar $30 \mu\text{m}$.

around 1-2 μm with the subset of size five. Taking into account the average cell-body diameter of 10 μm , I aimed to use a minimum of five scans thereon to register a gene expression pattern to a template.

Using the described procedure for registering gene expression patterns (cf. Gene expression pattern registration), I registered the expression patterns of 16 genes in 48 hpf, 31 genes in 72 hpf and 9 genes in 6 dpf templates.

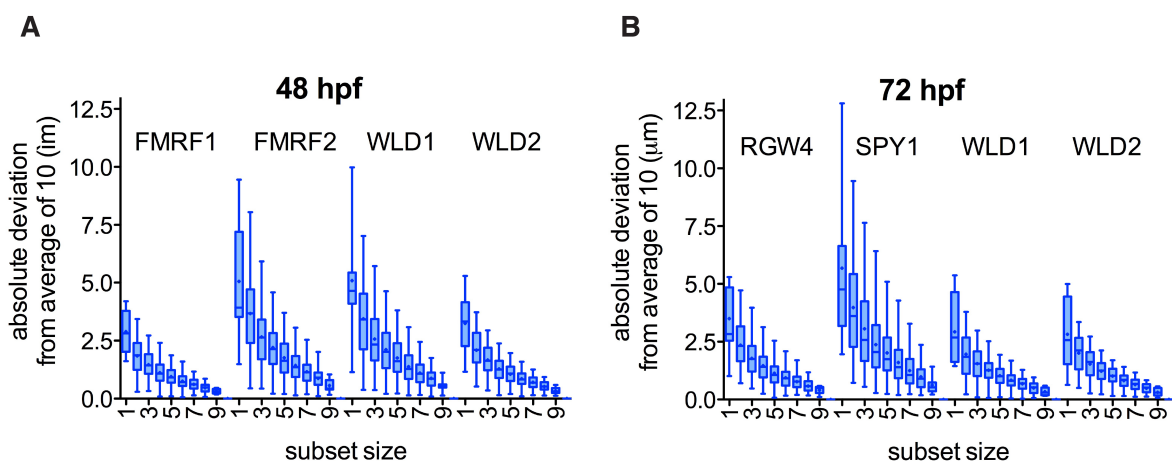


Figure 3.12.: Effect of sample size on average gene expression pattern.

For each cell, random subsets of different sizes (one to nine, 1000 iterations for each size) were sampled from the individual specimens. The absolute deviation of the cell center positions of each subset relative to the complete set average coordinate is calculated for 48 hpf (A) and 72 hpf (B) larvae and averaged over 1000 iterations. The cells and their identifiers are shown in Figure 3.11. The graphs represent min-max values, (+) indicates mean values. hpf, hours post fertilization.

3.3.4. Registration of immunostained specimens

Immunostained specimens reveal the neuronal processes of the gene expressing cells. Though such specimens undergo a different protocol compared to the *in situ* hybridization preparations, it is possible to register them to the *in situ*-based reference templates. To implement such a registration, the tubulin signal was registered to the generated DAPI-template and was further employed as a reference signal to register immunostained specimens. Visual inspection of the registered immunostained specimens revealed a close match of the corresponding anatomical structures in the immunostained samples and the tubulin-template (Figure 3.13 C).

3.3.5. Linking anatomical structure and function

Current gene labeling and imaging techniques localize anatomical regions associated with molecular activity. Knowing gene function, it is possible to imply functionality of the marked anatomical regions. Such strategy links animal anatomy and physiology. Gene expression atlases allow analysis of functionality of multiple genes in the context of anatomical structures. To provide a convenient platform for exploring animal physiology using the gene function information, I annotated the gene expression domains in the atlas with the corresponding function information obtained from the literature [32, 34, 33] or through in-house pharmacological experiments. Literature analysis revealed genes associated with functions such as: "increasing ciliary beating" (such genes as LY amide, L11, FMRF amide, SPY), "decreasing ciliary beating" (WLD), "phototaxis" (r-opsin-1) and "settlement" (myoinhibitory peptide, MIP).

Pharmacological behavioral experiments can be performed on *Platynereis* larvae to determine gene functionality. One of the strategies widely applied to *Platynereis* involves adding a neuropeptide to a cuvette with larvae (cf. Pharmacological experiments), where, after penetrating the larval bodies, the neuropeptide affects the animals in the same manner as an intrinsically generated neuropeptide. Hence, the observed effect, if any, can be contributed to that neuropeptide.

To quantify the effect of neuropeptides on locomotion, I developed a program that classifies and characterizes *Platynereis* locomotor activity from the video recordings of behavioral experiments. The program classifies the locomotion activity (swimming or crawling) and subse-

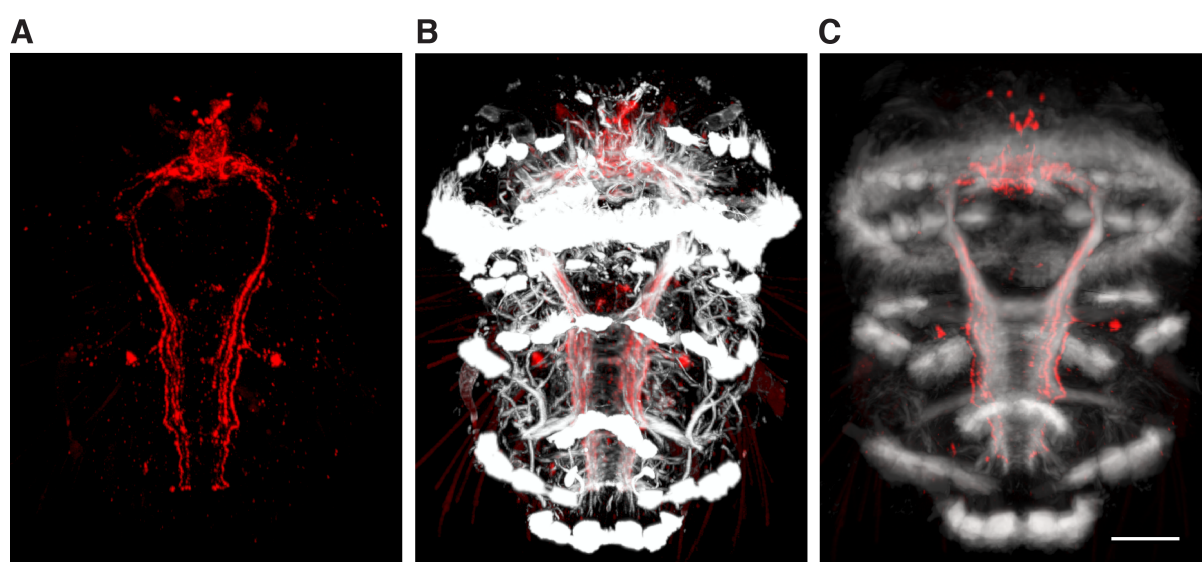


Figure 3.13.: Registration of immunostained specimens to the *in situ* hybridization template. (A) Expression pattern of FMRF amide (FMRFa) at 72 hpf. (B) Expression pattern of FMRFa (red) together with acetylated tubulin (white). (C) Registered expression pattern of FMRFa together with the tubulin reference signal. Scale bar 30 μ m. hpf, hours post fertilization.

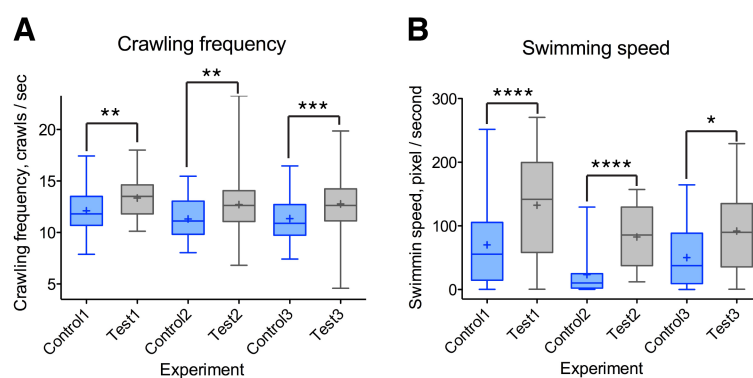


Figure 3.14.: Locomotion analysis of the FMRFa-treated larvae.

(A) Crawling frequency (crawls/sec) determined for the FMRFa-treated and control larvae in three repetitions. (B) Swimming speed (pixel/sec) determined for the FMRFa-treated and control larvae in three repetitions. hpf, hours post fertilization.

quently characterizes that activity (swimming and crawling speed or crawling frequency).

I evaluated the plugin by comparing its output to the manual classification on the test set of 10 individual larvae throughout 60 frames. Four instances ("instance" refers to an individual larva in a single frame) were wrongly classified by the plugin (0.7%). Most classification errors are attributed to the fact that the larvae were in a transitional state between swimming and crawling. I evaluated the frequency measurement based on a test set of five larvae that performed a combined total of 48 bends (Figure 3.7 C). The crawling frequencies determined by the program and the human expert were identical for three larvae and differed by less than 1.2% in two other larvae. I also compared the coincidence of bending events determined by the program and the human expert which revealed that 73% of bends were within the same frame or one frame apart, 17% of bends were two frames apart and 10% were three frames apart or more, or were not detected.

Here, I explored the effect of the neuropeptide FMRF amide (FMRFa) on 72 hpf larvae. The FMRFa expression pattern has previously revealed neuronal projections in the ventral nerve cord of the animal (Figure 3.13 B), which suggested that it influences *Platynereis* central pattern generator (CPG) and thereby affects swimming and crawling activities. To verify this hypothesis, the FMRFa neuropeptide was added in the cuvette with larvae and the larval behavioral response was video recorded. The developed plugin was then applied to measure the effect of the FMRFa neuropeptide on swimming and crawling speeds and crawling frequency of 72 hpf larvae. The results demonstrated that FMRFa significantly increases crawling frequency and swimming speed (Figure 3.14 E, F), whereas no significant effect on crawling speed was determined.

3.3.6. Interactive gene expression atlas

To provide a convenient representation to visualize and analyze gene expression patterns, I developed a visualization plugin for ImageJ that merges multiple channels into a single RGB image. However, the plugin is limited in the number of gene expression patterns that can be visualized simultaneously due to the image sizes.

As an alternative, I employed the 3D graphics software Blender for this purpose, due to its memory efficiency and a wide range of visualization, navigation and animation tools. I created a 3D model of the gene expression atlas, where widely expressed gene expression domains were represented as smoothed surfaces and cell-specific domains were approximated with spheres (Figure 3.4). To provide anatomical landmarks, I imported the averaged acetylated tubulin signal highlighting cilia and neurites.

The gene expression domains are handled as objects in the model, therefore they can be quickly located in the 3D view by gene name, or gene names can be identified for the domains of interest.

To provide components essential for the gene expression analysis, the model was extended using the Python scripting embedded in Blender. I implemented such options as querying the atlas by gene name (Figure 3.5 A, B), gene function (Figure 3.5 C) and gene type (Figure 3.5 D). It is therefore possible to display different combinations of gene expression patterns, e.g. genes increasing ciliary beating frequency or neuropeptides (Figure 3.5 B, C, D). Blender models were generated for three developmental stages: 48 hpf (Figure 3.15 A, B), 72 hpf (Figure 3.15C, D) and 6 dpf (Figure 3.15 D, E).

I provided the functionality to query the gene expression atlas for molecule colocalization. A user can specify two genes of interest and the overlapping volumes are then determined and displayed. Figure 3.16 shows the colocalization volumes for the AST-C and Deiodinase expression patterns and the gene coexpression matrix in the 72 hpf atlas.

3.3.7. Mobile and web applications for *Platynereis* gene expression atlases

The model of the *Platynereis* gene expression atlas was provided for public access via the On-line GeneExpressionAtlas (Figure 3.18) and Mobile GeneExpressionAtlas applications (Figure 3.17). An option to query expression domains by gene name was implemented (Figure 3.17 C, Figure 3.18 B). To speed up the applications on the devices with limited computational resources and improve user experience, the models were optimized by applying the *Decimate* modifier to each object, which reduced the number of the vertices in the model. The Three.js library [110] was used to visualize the 3D model in the web browser and the libGDX [107] library was employed for visualization on Android devices. The applications provided an efficient way to access the gene expression data without a need for installing 3D graphics software

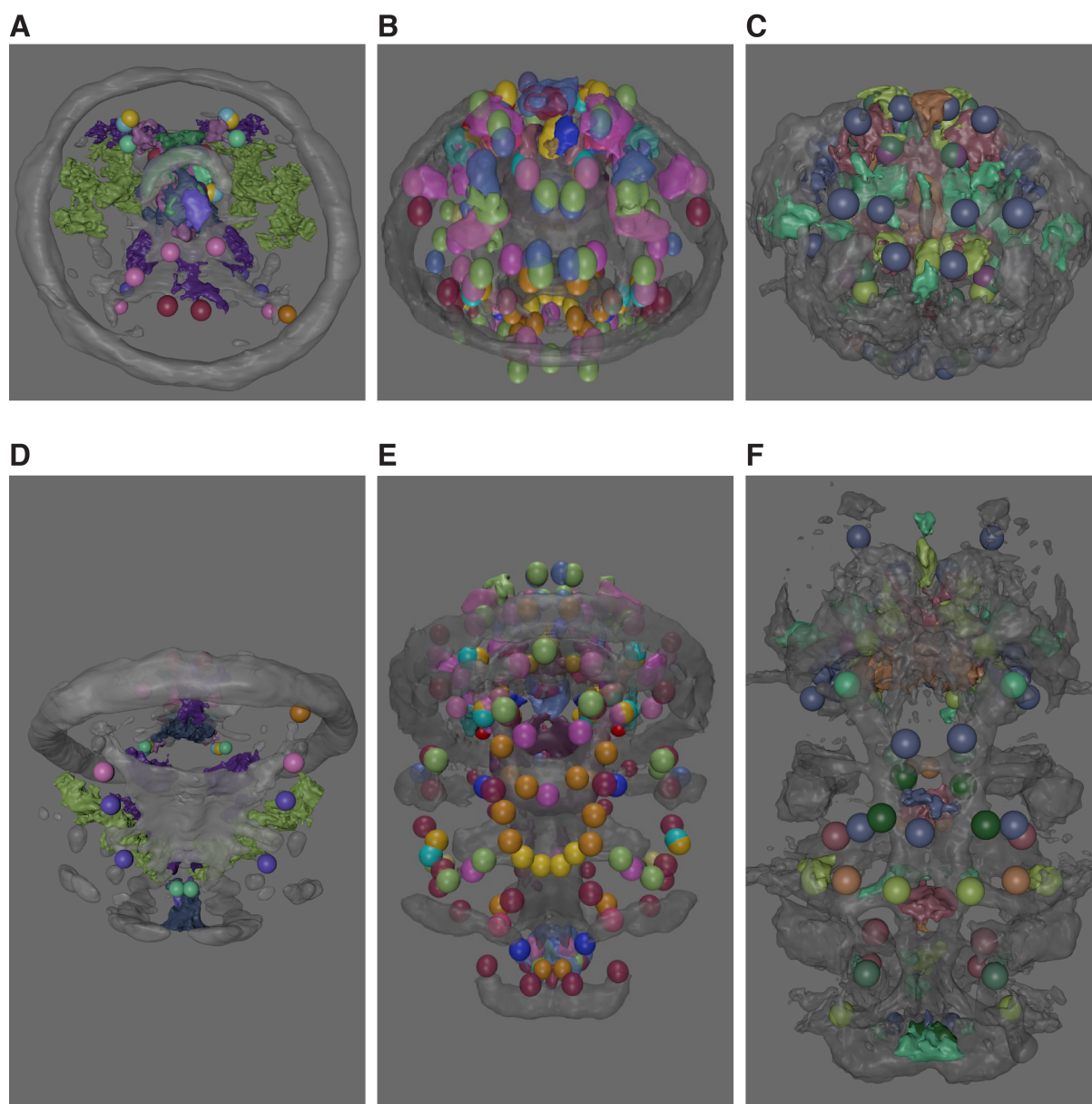


Figure 3.15.: Gene expression atlases modeled in 3D graphics software Blender. (A, D) The Blender model of the gene expression atlas at 48 hpf. (B, E) The Blender model of the gene expression atlas at 72 hpf. (C, F) The Blender model of the gene expression atlas at 6 dpf. (A-C) anterior views, (D-F) ventral views. Tubulin signal is shown in grey. hpf, hours post fertilization. dpf, days post fertilization.

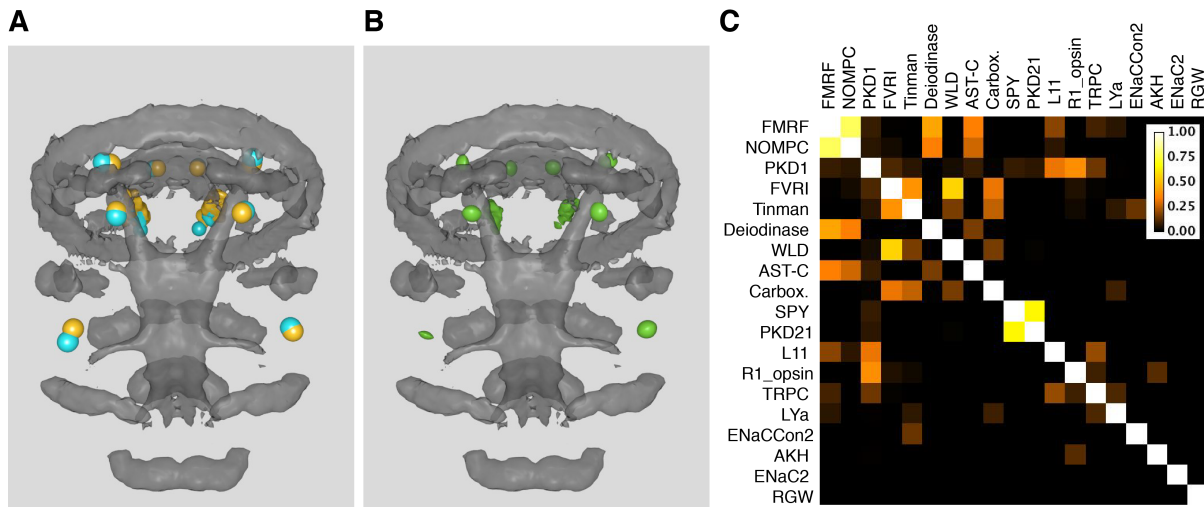


Figure 3.16.: Gene coexpression analysis in Blender.

(A) The AST-C (yellow) and Deiodinase (cyan) expression patterns in the 72 hpf atlas. (B) The coexpression volume for the AST-C and Deiodinase expression patterns. (C) The gene coexpression matrix in the 72 hpf atlas. hpf, hours post fertilization.

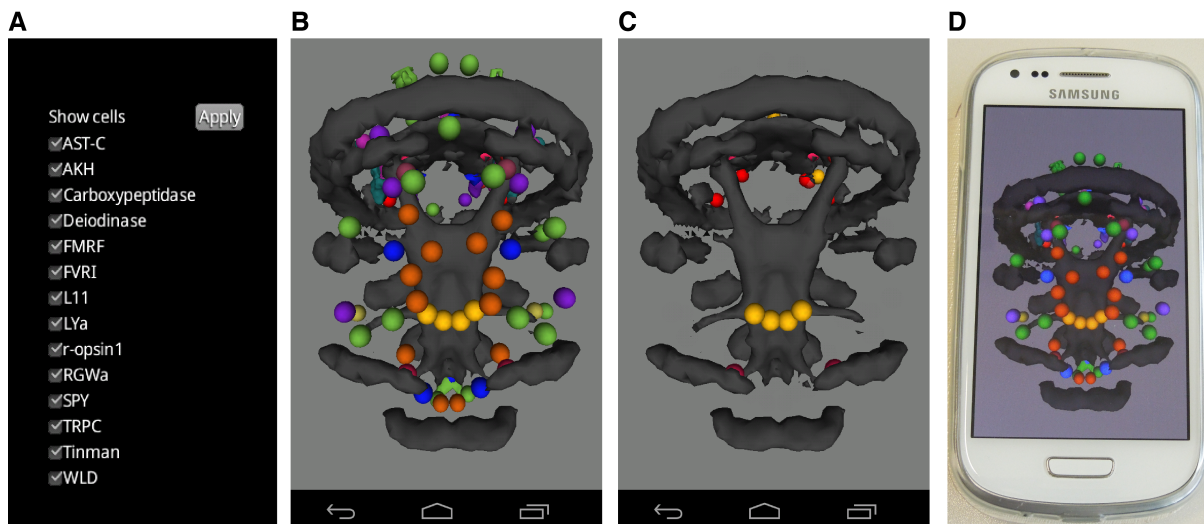


Figure 3.17.: Mobile GeneExpressionAtlas.

(A) The main menu for selecting gene expression patterns for visualization. (B, C) A screen for the interactive 3D visualization of the gene expression patterns selected in the main menu. (D) A smartphone running Mobile GeneExpressionAtlas.

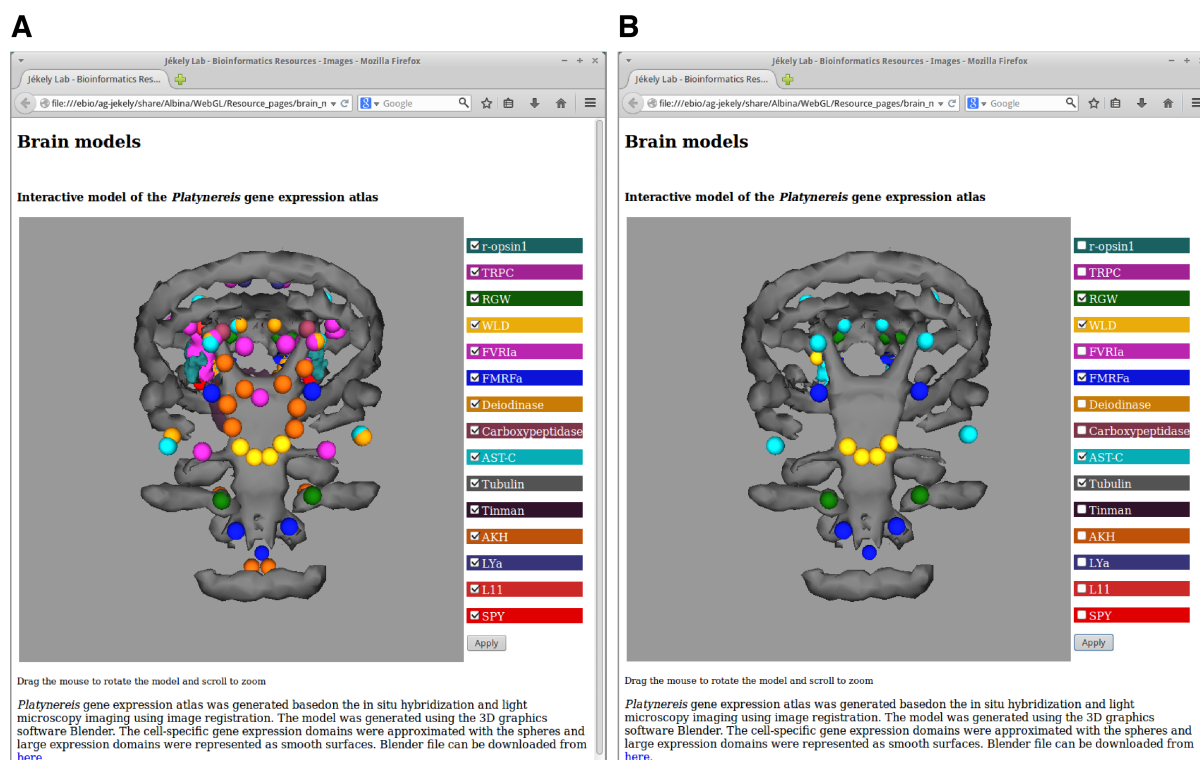


Figure 3.18.: Online GeneExpressionAtlas.

Screenshots of the Online GeneExpressionAtlas running in the web browser, which presents an interactive 3D view of a neuronal circuitry model and a list of gene expression patterns that can be selected for visualization. (A) All expression domains selected for visualization. (B) RGW (green), WLD (yellow), FMRF (blue) and AST-C (light blue) shown together with tubulin staining (gray).

or any knowledge of how to use 3D graphics software. The application gives a brief overview of gene expression data, which can be further explored by downloading the Blender model file.

3.3.8. Gene expression colocalization using gene expression atlas

The gene expression atlas can be used to determine colocalizing and therefore possibly interacting molecules. Visual inspection of the registered gene expression patterns at 72 hpf revealed broad overlap between the expression patterns of rhabdomeric opsin-1 (r-opsin-1) and TrpC (Figure 3.19 A-H). Subsequent double *in situ* hybridization (Figure 3.19 I-L) confirmed coexpression of the two genes in the ventral and dorsal eyes.

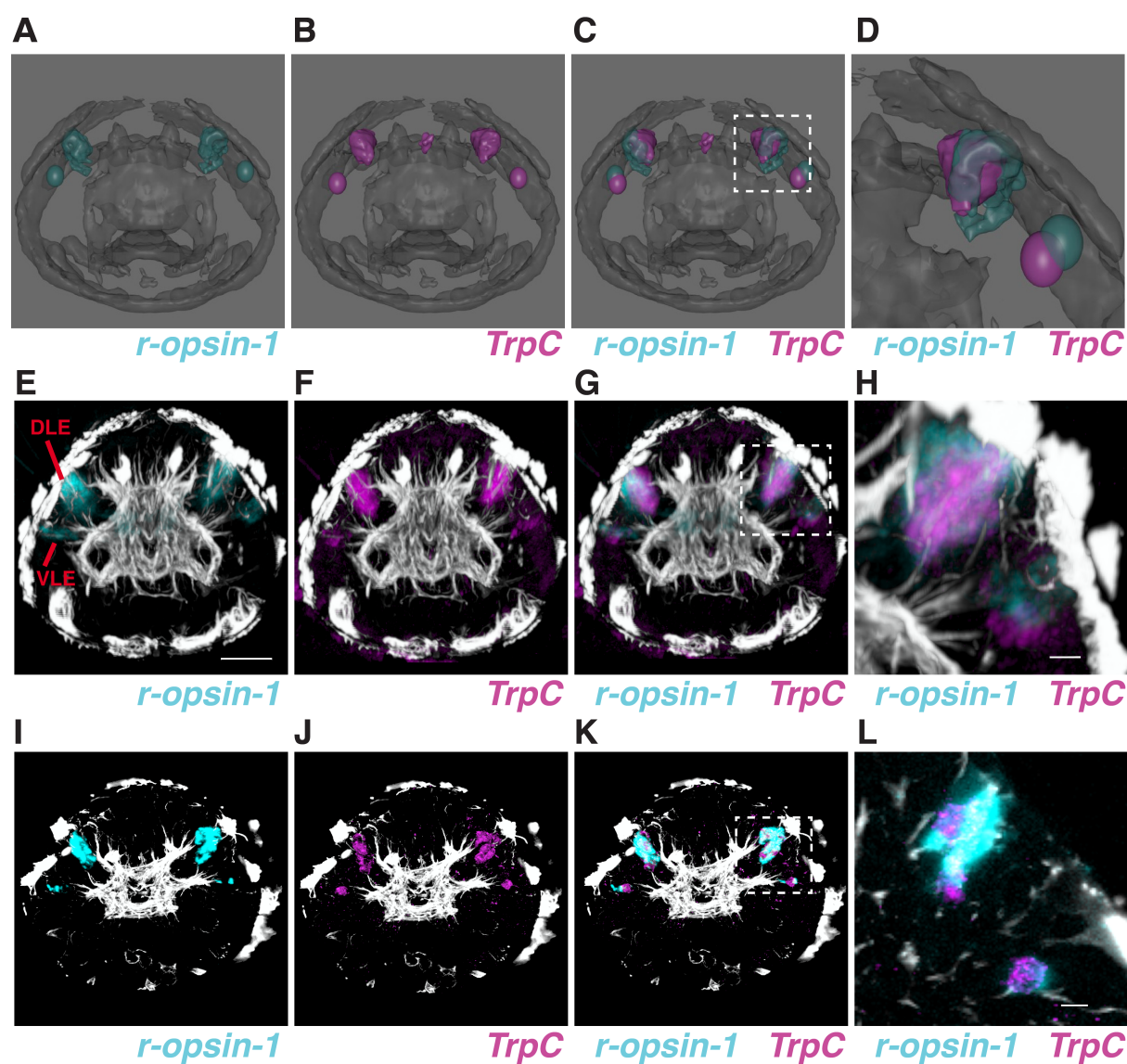


Figure 3.19.: Colocalization of TRPc and r-opsin-1 expression patterns. (A-D) Colocalization of the expression patterns of r-opsin-1 (cyan) and TrpC (magenta) at 72 hpf, as determined by the Blender model. (E-H) Colocalization of the expression patterns of r-opsin-1 (cyan) and TrpC (magenta) at 72 hpf, as determined in the registered images. (I-L) Colocalization of the expression patterns of r-opsin-1 (cyan) and TrpC (magenta) at 72 hpf, as determined by double *in situ* hybridization. Scale bar (E-G, I-K) 30 μm , (H, L) 10 μm . DLE, dorsal larval eye; VLE, ventral larval eye; hpf, hours post fertilization.

The atlas has been used to examine the mechanism of interaction of myoinhibitory peptide (MIP) and its receptor [33]. The registered expression patterns of MIP and its receptor were located in neighboring cells, thereby indicating the paracrine nature of this neuropeptide signaling pathway. Colocalization in the expression patterns of registered MIP and *dimmed*, a neuroendocrine cell marker in insects, further supported the neuroendocrine nature of MIP signaling.

The atlas can be used to efficiently identify potential gene coexpression, for example, the gene expression domains of AST-C (cyan), Deiodinase (yellow), FVRI (pink) overlapping with each other in the ventral body region (Figure 3.5 B). Such candidates for coexpression can be verified by performing double *in situ* hybridization, a method that labels two gene expression patterns in the same specimen, thereby revealing the exact relative distribution of the gene expression patterns.

3.4. Discussion

The current work demonstrated that image registration can be successfully applied to *Platynereis* images to generate a whole-body gene expression atlas for the early developmental stages. The atlas describes the molecular expression signature for different anatomical regions and provides a link between those regions and corresponding physiological activities. The user-friendly representation of the atlas incorporates an efficient means of visualizing and analyzing the gene expression patterns and provides a convenient platform for integrating and accessing various attributes associated with a gene. Such representation greatly facilitates data exploration and has already proven useful in solving practical problems such as detecting candidate genes for coexpression, understanding gene interaction mechanisms, and relating behavioral information to anatomical structures.

3.4.1. Image registration protocol

Image registration The image registration protocol implemented here was extensively evaluated and was found suitable for registering images of the early developmental stages of *Platynereis*. First, rigid registration was examined. The evaluation revealed that 85-90% of images are registered correctly, when using the landmark-based method for 48 hpf images and the intensity-based method for 72 hpf and 6 dpf images. The intensity-based method demonstrates low performance on the 48 hpf larvae, which was due to the spherical shape of the 48 hpf larvae. In contrast, the elongated body of the 72 hpf and 6 dpf larvae facilitates the intensity-based registration.

Non-rigid (affine and deformable) registration was also assessed, using the image similarity

metric (Mattes mutual information metric) and resulted in 95-100% of images registered with a registration match of 60% or higher. The evaluation of the image registration protocol demonstrated that it can be reliably applied to the images of *Platynereis* larvae.

Template generation The image registration procedure was iteratively applied to generate average anatomical templates for the early developmental stages of *Platynereis*. The original iterative strategy for generating a template [53, 88] was biased towards the image selected in the first iteration [81], which might have unusual morphological features compared to the majority of the population. The bias-compensation procedure was introduced in the protocol to achieve subject-independent templates.

Before mapping gene expression patterns to the templates, the accuracy of the registration procedure was evaluated based on the absolute deviation of the corresponding cells in individual specimens. The absolute deviation was in general lower than $10\ \mu\text{m}$ (comparable to cell diameter), thereby demonstrating a near-cellular resolution of the gene expression pattern registration. Low absolute deviation values also indicated high synchrony of *Platynereis* early development and showed that to a large extent the registration protocol overcomes technical and natural variation.

Absolute deviation was also employed to compare nuclear stain and acetylated tubulin as reference markers for the template generation. This parameter showed that the nuclear-stained templates often outperform the tubulin templates, which can be attributed to the fact that the nuclear staining provides a non-homogenous signal for registration over the entire larval body. The nuclear-stained templates were further utilized for registering gene expression patterns in *Platynereis* at the early developmental stages.

3.4.2. Comparison with VIB, ISA and PrImR protocols

Frequently, atlas generation methods are tailored specifically for the organism under study and are not applicable to different organisms or different data formats. According to the literature, virtual insect brain (VIB) [80] and iterative shape-averaging (ISA) [88] protocols, as well as the current method, belong to the species- and data-independent protocols. The VIB protocol, originally employed to generate an anatomical atlas for the *Drosophila* brain, was primarily developed for volumetric analysis and therefore preserves the volumes of the brain regions. It requires prior segmentation of the anatomical structures, which guides the subsequent registration. Average templates based on VIB protocol reflect the volume variability in populations.

The aim of this project was to generate a standard average anatomy and a representative gene expression map, therefore the protocol aims at reflecting an average morphology and removing individual variability, in contrast to the VIB protocol. The presented method is automatic and does not require any pre-processing steps, which makes its usage more straightforward. The

current protocol is more similar to the ISA protocol, which also aims at generating an average atlas by removing individual variation. Both protocols subsequently apply rigid, affine and non-rigid transformation for registration and employ an iterative procedure to generate a template. The main difference of the current protocol is the bias-compensation procedure. Jundi *et al.* compared the VIB and the ISA protocols, and concluded that both protocols are biased towards the image selected as a first template [81]. That image introduces a bias in terms of the positions of the anatomical structures in the VIB template and in terms of the volumes of the anatomical structures in the ISA template.

The presented registration protocol extends the PrImR protocol [53], which employed ITK-implemented registration and was applied to the episphere of 48 hpf larvae. Here, I applied image registration to the whole-body scans, enabled by TDE as a mounting medium for microscopy, and provided an extensive overview of the *Platynereis* anatomy and a wide range of gene expression patterns. More robust rigid registration methods were introduced in the protocol, which were shown to be suitable for the whole-body images. The stereotypic development of *Platynereis* enabled the generation of the gene expression atlases for the three developmental stages and the consequent tracking of the temporal dynamics of gene expression patterns.

3.4.3. Nuclear stains as universal reference markers

Various reference markers have been employed for generating anatomical atlases [76, 53, 77, 79, 81, 86]. The acetylated-tubulin signal was utilized earlier in *Platynereis* as a reference marker to generate the template and subsequently to register gene expression patterns [53]. The results presented here demonstrated that DAPI stain can be used as a reliable reference marker for generating a template for *Platynereis* larvae. Nuclear stain has also been utilized in the recently published registration protocol for zebrafish larvae [78]. Here, I propose nuclear stains as universal reference markers for generating anatomical atlases. In contrast to antibody markers, such stains give the broadest possible label of anatomy by labeling every cell, whereas the signal of antibody markers can be absent from large regions of the body and consequently provide less information for registration (e.g. acetylated tubulin signal). The nuclear-stain templates provide a fuller representation of anatomy and yet the provided signal is not homogeneous and reveals detailed internal structure (for example, it delineates neuropil, nerve cord and muscle structures in *Platynereis*). Nuclear stain is cheap, easy to perform, and compatible with standard *in situ* hybridization protocols. Due to the aforementioned points, I strongly recommend nuclear stains as the first markers of choice for future atlas generation projects.

3.4.4. Immunochemistry for behavior-regulating neuropeptides

In previous reports [76, 88, 89, 96], immunostained specimens have been successfully registered to anatomical templates and the registered expression patterns have been applied for exploring neuronal networks. Such registration revealed structural organization of the nervous system on the level of single cells [89]. It allowed analyzing neuronal connectivity, formulating hypotheses on neural pathways and the identification of novel physiological properties of particular cell types. Combined with physiological measurements, immunochemistry data provided insights into neuronal circuitry function [88].

Here, *Platynereis* specimens immunostained with antibodies against the FMRF amide neuropeptide (FMRFa) were registered to an average anatomy template. FMRFa staining revealed neuronal projections in the nerve cord, which led to the hypothesis that FMRFa influences the central pattern generator (CPG) of *Platynereis* locomotion. Behavioral experiments on *Platynereis* locomotion confirmed that FMRFa increases crawling frequency and swimming speed, thereby supporting the connection between the FMRFa-expressing cells and CPG locomotion.

Exploring gene expression patterns in combination with behavioral experiments provided a unique opportunity to relate anatomical regions to their corresponding function, which is the main goal of anatomical and physiological studies. To provide extensive data for gene exploration, the gene expression atlas was extended with functional and behavioral information obtained from additional behavioral experiments and the literature review.

3.4.5. Interactive model of the gene expression atlas

The image registration protocol outputs average gene expression patterns in the form of image stacks, therefore handling and simultaneously visualizing multiple gene expression patterns is a memory-intensive task. An efficient solution to this problem was provided by the 3D graphics software Blender, which encompasses versatile tools for visualizing and navigation through data. I created a model of the gene expression atlas in Blender, which also integrates additional information associated with genes. The model can be further adapted to particular research needs via embedded Python scripting. Here, I provided the functionality and the interface for exploring gene expression patterns, such as querying expression patterns by gene name, type or function.

The model facilitated analyzing genes for coexpression and understanding molecular mechanisms. It revealed colocalization of the expression patterns of TRP-C and rhabdomeric opsin-1 in the adult eyes of the 72 hpf *Platynereis* larvae, which was subsequently confirmed by double *in situ* hybridization. This finding suggested that *Platynereis* photoreceptors might employ a phototransduction cascade similar to that of insect rhabdomeric photoreceptors [114], strength-

ening the idea that these photoreceptors represent a homologous cell type [115].

The disadvantage of the Blender representation is the loss of the original image intensities when generating the gene domain surfaces. Therefore, candidates for coexpression determined in the Blender model should be verified by inspecting the original gene expression patterns or other experimental procedures (i.e. double *in situ* hybridization).

The model of the *Platynereis* atlas was inspired by the Virtual worm project [116], which represents a cell-by-cell reconstruction of an adult nematode *C. elegans*. The aim of the Virtual worm project was to generate an interactive atlas of the *C. elegans* for learning its anatomy.

In contrast, the model designed in this work concentrates on gene expression domains, rather than individual cells. A model focusing on individual cells would represent a more rigorous strategy. It would require resolving single cells in the entire body (e.g. in a DAPI-stained specimens) and then annotating the cells with the expressed genes. However, the available techniques did not reliably resolve individual cells in the imaged *Platynereis* specimens, therefore the domain-based approach was used.

The Virtual worm also visualizes *C. elegans* tissues and organs. The model of the *Platynereis* gene expression atlas can also be extended with further anatomical information, using tissue- or organ-specific markers (such as phalloidin staining of muscles or r-opsin markers staining photoreceptors), which could empower exploring gene expression patterns in combination with anatomical information. In contrast to the Virtual worm depicting an adult worm, the *Platynereis* model is restricted to the early developmental stages, due to the restriction of confocal microscopy to a smaller specimen.

3.4.6. Gene expression atlas in *Platynereis* research

The image registration protocol implemented here has been engaged in several studies involving *Platynereis*, for example, to explore mechanisms of *Platynereis* larval settlement [33], vertical migration [32], and phototaxis [12]. The gene expression map was employed to explore mechanisms of molecular interactions and suggested a paracrine signaling between myoinhibitory peptide (MIP), a neuropeptide involved in *Platynereis* settlement, and its receptor [33]. The image registration protocol was also used for mapping the neuropeptides involved in larval vertical migration, which revealed largely non-overlapping expression domains of these peptides [32]. The atlas elucidated genes that are expressed in the adult eyes, thereby revealing the candidate molecules involved in *Platynereis* vision [12].

3.4.7. Future directions

The gene expression atlas embodies a snapshot of the molecular complexity of the *Platynereis* larval anatomy and establishes types and molecular profiles of individual cells. It yields insights

into the detailed structural and functional organization of anatomical regions and provides the basis for precise genetic and functional manipulations, such as guiding the selection of genes for expression manipulation experiments or physical stimulation and cell ablations.

The atlas can be used to reveal inter-individual differences in gene expression patterns and to relate these differences to variations in anatomy or behavior, thus making the atlas helpful for phenotyping and molecular characterization of subpopulations (such as genetically or physiologically manipulated specimens). To identify population-specific characteristics, individual specimens can be compared to the standard atlas. Individual specimens can also be compared to each other after being normalized to the standard atlas. Additionally, the gene expression atlas can be used for molecular characterization of new specimens and can even be integrated in an imaging process to automatically detect cells for stimulation or ablation.

Extending the gene expression atlas with further gene expression patterns will provide a more comprehensive map of spatial gene distribution. The atlas can be expanded to other stages of *Platynereis* development in order to follow temporal dynamics of gene expression patterns. To make the atlas accessible for the community and stimulate community input to its further development, the average anatomy templates, instructions and image registration scripts were provided for public access. The atlas presented here will be a useful tool for guiding future projects involving *Platynereis*.

4. Neuronal circuitry model for *Platynereis dumerilii*

4.1. Introduction

Organisms continuously receive signals from the environment via their sensory organs. Environmental signals are further processed in the nervous system to generate an appropriate response. Neuroscience explores the flow of information within neuronal circuits and the mechanisms that modulate the response reaction.

Imaging and image analysis techniques are widely used in the study of neuronal circuits. One imaging technique, electron microscopy, resolves biological structures at a very fine resolution, thereby elucidating the morphology of individual cells and even of subcellular structures. Electron microscopy, therefore, captures individual neuronal processes and synapses in the context of neuronal circuitry. Using designated image analysis software, neuronal circuitry elements can then be reconstructed from electron microscopy images and studied to elucidate network function.

4.1.1. Nervous system reconstructions in model organisms

In 1986, White *et al.* [117] reconstructed the neuronal circuitry of the nematode *Caenorhabditis elegans* (*C.elegans*) from electron microscopy (EM) images and described the structure and the connectivity of the *C.elegans* nervous system. This was the first incidence of EM reconstruction of a whole organism. The authors characterized the organization of the neuronal processes, described the connectivity patterns of the nematode and classified neurons based on morphology and connectivity. The reconstructed *C.elegans* network was analyzed in several subsequent studies [118, 119, 120, 121, 122]. Varshney *et al.* [120] extensively explored statistical and topological properties of the *C.elegans* neuronal network and discovered structural similarities to the mammalian neuronal cortex. Jarrell *et al.* [118] focused on the neural substrate responsible for *C.elegans* mating behavior, classified the involved neurons and determined their functions. Using network partitioning, they determined distinct modules regulating

different basic behaviors, such as searching for vulva, locomotion, posture and insemination. Sohn *et al.* [122] also employed topological clustering methods to determine structural modules in the entire *C.elegans* neuronal network and defined the functionality associated with the modules. Bumbarger *et al.* [119] compared the *C.elegans* pharynx neuronal organization to that of another nematode, *Pristionchus pacificus*, and discovered significant rewiring of the synaptic connectivity in the pharynx between the two species, despite the fact that the neuron types were conserved.

The neuronal network of another model organism, *Drosophila melanogaster*, has been explored using EM [11, 123, 124]. Cardona *et al.* [123] reconstructed five compartments of the *Drosophila* larval brain, classified the reconstructed neurites based on morphology and determined the most common motifs in the network. The authors also determined statistical properties of the network and compared them to those of the mammalian cortex. Another study of the nervous system of *Drosophila* combined EM data with functional imaging to study direction-selective ganglion cells responsible for visual motion detection [11]. The authors attributed the directional selectivity to the asymmetry in the structure and the connectivity determined from the EM reconstructions. A later study by Takemura *et al.* [124] was also concerned with directional selectivity in *Drosophila* and provided a more comprehensive reconstruction of the *Drosophila* optic medulla. The authors matched the EM reconstructions to the light microscopy data and consequently identified the types of neurons responsible for motion detection.

Additionally, EM reconstructions were used to investigate vision in mammals [125, 126, 127]. Bock *et al.* [125] investigated physiology and neuronal anatomy in the mouse visual cortex. The authors determined direction-selective neurons using calcium imaging and characterized those neurons using EM reconstructions. Helmstaedter *et al.* [127] investigated the neuropil region in the mouse retina. The authors classified the reconstructed neurons based on connectivity type and determined a new interneuron type based on their classification. Song *et al.* [126] analyzed synaptic connectivity properties of the rat visual cortex and revealed several non-random features of the neuronal network.

4.1.2. Software for electron microscopy images

Using electron microscopy for resolving neuronal anatomy yields large volumes of data, therefore specialized software is employed to efficiently handle the acquired EM images. These software tools are also used to subsequently montage and align images. Due to the size of a specimen, an individual tissue layer might span multiple image tiles, hence, the obtained tiles should be montaged into a single image. The resulting montaged layers then need to be aligned into a stack to compensate for the distortions introduced during sectioning and imaging. Soft-

Software	Alignment	Segmentation	Web-based	3D view	2D graph
Catmaid [131]	no	yes	yes	yes	no
ConnectomeExplorer [132]	no	yes	no	yes	yes
Elegance [133]	no	yes	no	no	no
Espina [134]	no	yes	no	yes	no
Eyewire [135]	no	yes	yes	no	no
IMOD [128]	yes	yes	no	yes	no
Knossos [136]	no	yes	no	yes	no
Neurotrace [137]	no	yes	no	yes	no
Reconstruct [130]	yes	yes	no	yes	no
Rhoana: Mojo/Dojo [138]	no	yes	no	yes	no
Sopnet [139]	no	yes	no	no	no
TrakEM2 [129]	yes	yes	no	yes	no
Viking [140]	no	yes	yes	yes	yes

Table 4.1.: Software for processing and analyzing electron microscopy data.

ware applications such as IMOD [128], TrakEM2 [129] or Reconstruct [130] (Table 4.1) are capable of montaging and aligning EM images.

The aligned images can be further segmented to reconstruct neuronal processes and annotate synapses. Several attempts have been undertaken to automatically reconstruct neuronal processes. Machine-learning approaches were employed to segment EM images based on artificial neural networks [141], convolutional networks [142, 143], adaptive boosting [144], and random forests [145, 146, 147, 148]. Ridge detection and subsequent region merging was employed to detect neuronal membranes [149, 150, 151]. O’Reilly *et al.* [152] additionally employed a shape prior (circle) to segment membranes in individual layers. Operating on supervoxels rather than individual voxels was shown to be more robust towards local noise and reduced the computational complexity of the segmentation task. Graph partitioning [153], graph cut [154] and random forest [155] were applied to supervoxels to segment the neurons from the EM images. Segmentation can be further improved by integrating information from the adjacent sections as a geometrical constraint for graph-cut optimization [146, 147, 156] or level sets [157].

Automatic segmentation methods are often tailored for datasets with certain sample preparation methods, scanning technique or image quality. Being tested only on high-quality EM images, some methods are not applicable to the real-world noisy or disrupted data [158]. Another problem is that segmentation methods allow for a certain level of failure, thus, subsequent manual proofreading of the automatically segmented data is needed to achieve accurate results. This can be time-consuming and labor-intensive. Due to these difficulties, many researchers still employ manual or semi-automatic approaches for network reconstructions.

Several applications have been developed for segmenting and reconstructing EM images (Table 4.1). While some of the developed tools are not available or no longer supported, other tools are widely used by the community, for example, Knossos or TrakEM2. TrakEM2 was employed to reconstruct EM images of *Drosophila* [123, 159, 160, 161], the marine annelid *Platynereis dumerilii* [12], the nematode *Pristionchus pacificus* [119], the acoel *Symsagittifera roscoffensis* [162] and the mouse *Mus musculus* [124]. Knossos was also used for reconstructing mouse retina [11, 127].

The reconstructed neuronal processes must subsequently be analyzed, a significantly challenging task due to the complexity of neural system organization. Various tools have been developed for extracting information from reconstructed neuronal networks. The spatial overview of the neuronal anatomy can be obtained via 3D visualization tools provided within some of the afore-mentioned software, for example, 3D Viewer in TrakEM2, 3D Viewport in Knossos or Structure Viz in Viking. Neuronal reconstructions can also be exported and explored using external tools for 3D visualization, such as the 3D graphics software Maya [163], Blender [164] or Amira [165]. However, 3D representation disregards network connectivity information and, thus, hinders the analysis of information flow within the neuronal network.

To explore network connectivity, a reconstructed neuronal network can be represented as a graph where neurons represent nodes and synapses represent edges. Some of the afore-mentioned applications for reconstruction can provide 2D connectivity graph viewers, for example, Circuit Viz in Viking or Graph View in ConnectomeExplorer. Synaptic connectivity can also be exported and analyzed using some of the numerous tools for graph exploration, such as Gephi[166], Graphviz [167], Brain Connectivity toolbox [168] for Matlab, python libraries (NetworkX [169], igraph [170], graph-tool [171]), and others. Nevertheless, such graph representations do not allow for the projection of synaptic connectivity in a 3D anatomically relevant context, which is also important for understanding a network.

4.1.3. A new software tool for neuronal network exploration

In this study, I present NeuroDetective, software that integrates spatial and connectivity information and provides tools for analyzing information exchange between neuronal units. It provides a convenient platform for visualization and analysis of neuronal networks, thereby enhancing their exploration. NeuroDetective explores information flow within neuronal networks via tools for querying synaptic connectivity. I implemented the functionality to explore statistical properties of synaptic connectivity and calculate various network properties. NeuroDetective stores and accesses additional attributes for a neuronal circuit, such as gene expression, behavior and anatomical classification, thereby providing a knowledge base for the

network exploration.

I employed NeuroDetective to analyze neuronal circuitry of the marine annelid *Platynereis*. The relatively simple organization of the *Platynereis* nervous system at the early developmental stages despite complex behaviors makes it an attractive system for investigating neuronal circuitry using electron microscopy. *Platynereis* cell types share common ancestry with both vertebrates and invertebrates [43], therefore comparative studies of *Platynereis* neuronal circuitry can grant insights to the evolution of neuronal circuitries responsible for certain behaviors.

In this project, I focus mainly on the neuronal circuitry downstream of the adult eyes in order to understand the neuronal processes modulating vision-mediated behavior in *Platynereis*. Both the adult and the larval eyes are responsible for phototaxis in *Platynereis* [71, 12], which is a crucial behavior for light-mediated vertical distribution, larva dispersal, predatory avoidance and minimization of exposure to UV radiation [172]. The *Platynereis* eyes have already been studied from evolutionary, comparative and molecular perspectives [68, 71, 173, 174]. Arendt *et al.* [173] investigated *Platynereis* larval and adult eyes based on the expression of eye-specific genes, and located genes shared among the eyes of protostomes. Backfish *et al.* [68] molecularly characterized the photoreceptors in *Platynereis* and revealed rhabdomeric opsins (r-opsins) orthologous to those in zebrafish. Randel *et al.* [174] identified novel r-opsins in *Platynereis* eyespots and characterized their spatial and temporal expression. Jékely *et al.* [71] studied the role of the larval eyes in *Platynereis* phototaxis at the early larval stages.

Here, I concentrate on the neuronal basis of *Platynereis* late larval phototactic behavior. I applied NeuroDetective to explore the reconstructed neuronal circuitry of the 72 hours post fertilization (hpf) *Platynereis* larva in order to understand the flow of the electrochemical activity within the reconstructed neuronal circuitry and to unveil neuronal mechanisms modulating *Platynereis* behavior.

I also developed plugins to analyze behavioral experiments performed to verify hypotheses originating from the analysis of the reconstructed neuronal circuitry. Furthermore, I created a web application for interactive exploration of neuronal circuitries requiring only a web browser with enabled WebGL. I also provided an Android application, which allows the exploration of neuronal networks offline on mobile devices.

The developed tools enable the analysis of the biological information flow within a nervous system and improve our understanding of mechanisms of interactions between neuronal units. This information is crucial for realizing the proper capacities of an organism. Analyzing the network using the developed software sheds new light on the evolution and development of the nervous system of *Platynereis* and its functional capabilities.

4.2. Methods

4.2.1. Electron microscopy

The 72 hpf *Platynereis* larvae were prepared and imaged as described in [174]. In short, the larvae were fixed using high-pressure freezing with free-substitution, epon-embedded and sectioned (about 50 nm). Individual sections were contrasted with uranyl acetate and lead citrate and stabilized with carbon coating. The prepared sections were scanned on a transmission electron microscope (TECNAI Spirit) at a pixel size at 3.72 nm. Each section was scanned as a sequence of overlapping tiles. Electron microscopy was performed by Nadine Randel and Reza Shahidi.

4.2.2. Neuronal network reconstruction

The transmission electron microscopic images were imported into TrakEM2 [129], a Fiji [101] plugin that stitches image tiles from an individual layer and aligns layers to each other. TrakEM2 was also used to trace individual cells and to detect synapses. Two approaches were employed to trace cells: using AreaList, which represents objects as a set of separated areas, or AreaTree, which represents objects as a branching skeleton (a set of connected nodes). A unique ID was automatically assigned to each reconstructed object. Image alignment was performed by Nadine Randel, segmentation was performed by Nadine Randel, Csaba Veraszto, Luis Bezares and other members of the Jékely group.

4.2.3. Neuronal network model

The morphological structures reconstructed in TrakEM2 were exported as surfaces (AreaLists) or skeletons (AreaTrees) in OBJ format using the Fiji plugin 3D Viewer [175]. The neurons were imported into Blender using manual and automatic approaches. In Blender, cell bodies were then approximated with spheres and axons and dendrites with tubes (Figure 4.1 D, E). In the manual approach, the objects were hand-painted in the 3D view closely following the original traces. This method was primarily used for the neurons in the AreaList form.

The automatic method was used for the neurons in the AreaTree form. This method involved parsing OBJ files using the Python API implemented in Blender (Figure 4.2 E). Information on skeleton nodes and edges contained in an OBJ file was loaded into a graph structure (using a *dict* class in Python). The branches were traversed using a depth-first search (DFS) graph traversal algorithm and terminal branches shorter than a given threshold were eliminated. The remaining branches were then smoothed and converted into NURBS or Bezier curves in Blender. The cell

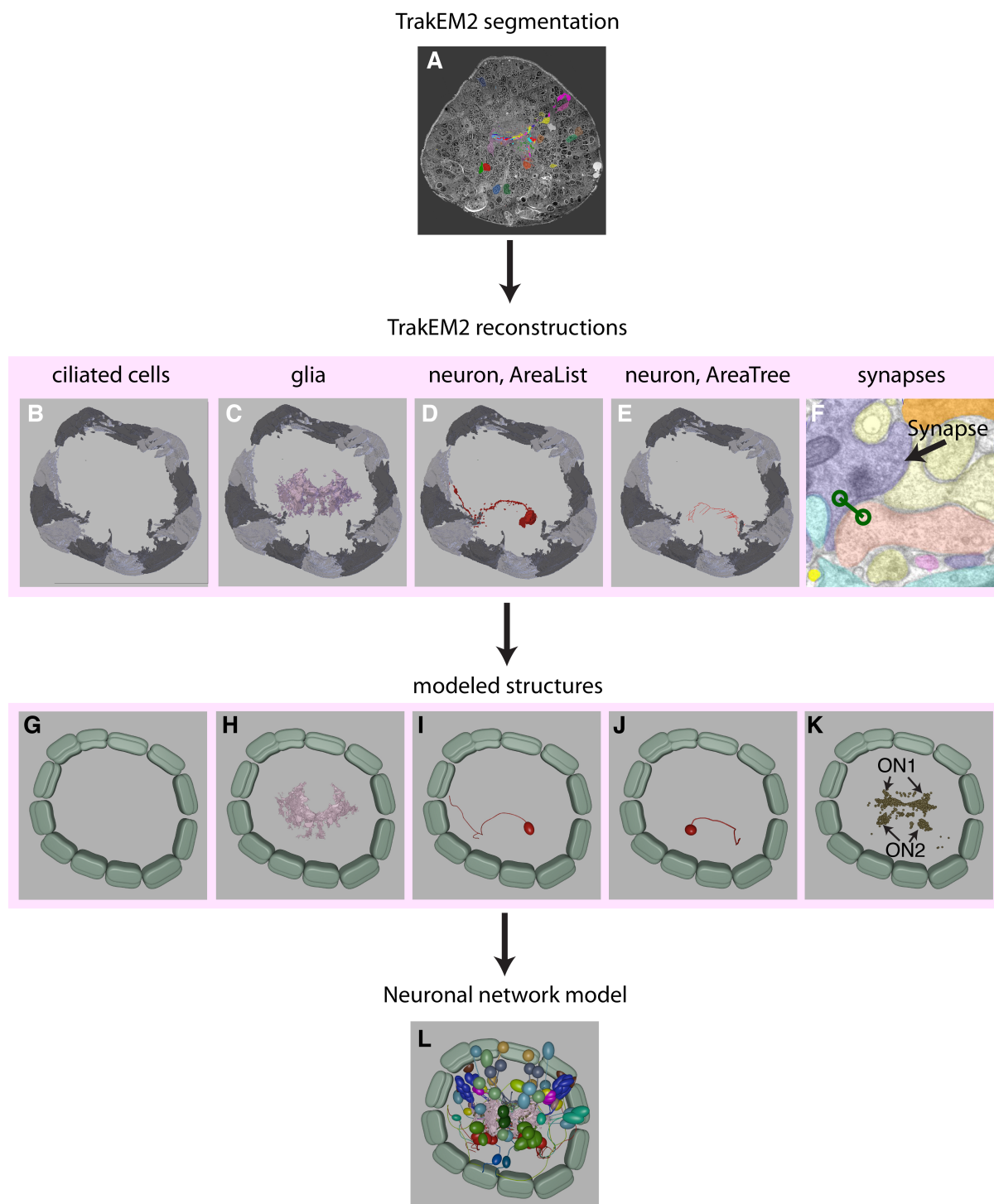


Figure 4.1.: A pipeline for creating a 3D model of a neuronal network.

(A) An individual electron microscopy image segmented in TrakEM2. (B, G) Ciliary band cells segmented in TrakEM2 (B) and modeled in Blender (G). (C, H) Glia cells segmented in TrakEM2 (C) and imported in Blender (H). (D, I) An individual neuron segmented as AreaList in TrakEM2 (D) and modeled manually in Blender (I). (E, J) An individual neuron segmented as AreaTree in TrakEM2 (E) and imported automatically in Blender (J). (F) An individual synapse annotated in TrakEM2 and (K) all synapses imported automatically in Blender. (L) Final neuronal circuitry model showing all neurons and synapses of the reconstructed neuronal circuitry in *Platynereis*. ON, optic neuropile.

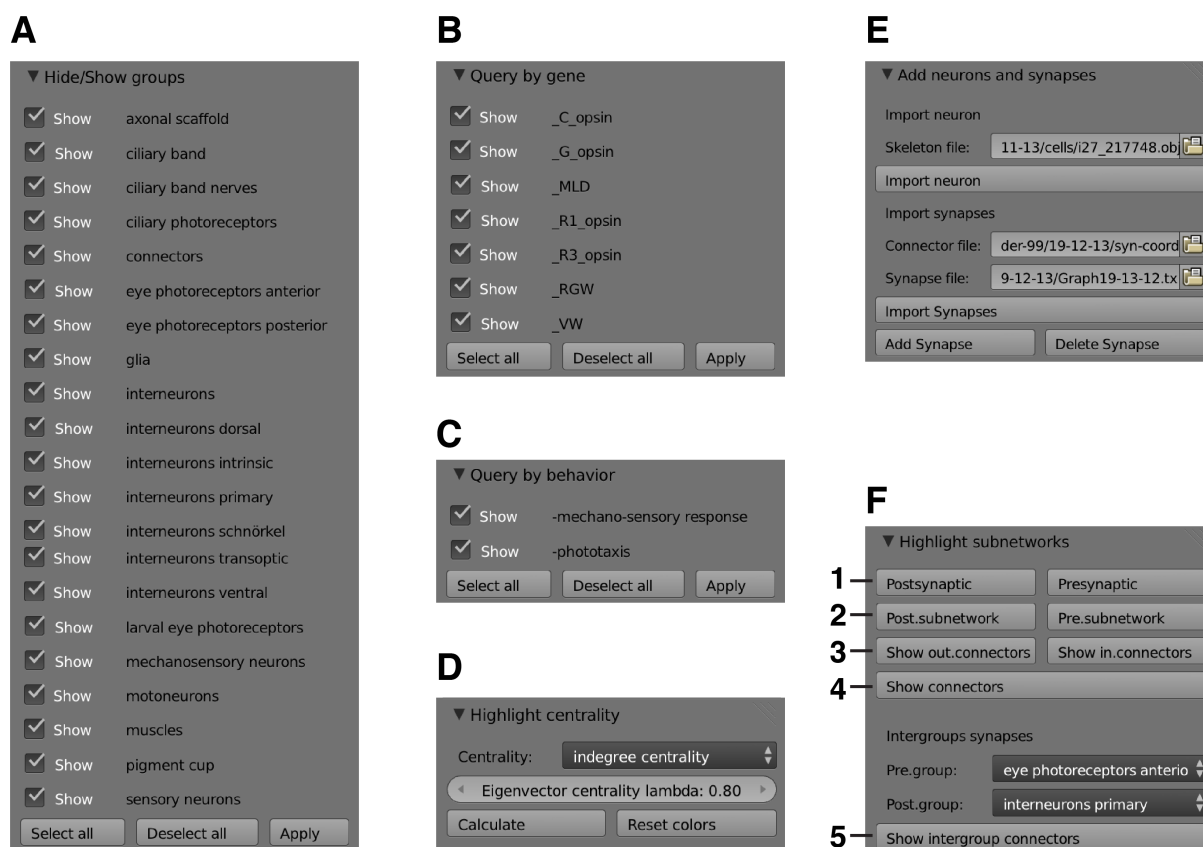


Figure 4.2.: NeuroDetective user interface.

(A, B, C) Panels for highlighting neuronal network elements based on anatomical classification (A), gene expression (B) or behavior (C). (D) A panel for calculating and visualizing centrality measures for the elements of a neuronal network. (E) A panel for automatically importing neurons and synapses. (F) A panel for exploring synaptic connectivity for querying all pre- and postsynaptic sites for a selected cell (1), highlighting up- or downstream subnetworks (2), displaying all incoming or outgoing synapses for the selected object (3) or visualizing all synapses between two cells (4) or two anatomical groups (5).

body was approximated with a sphere placed in the location of the starting node of a skeleton (Figure 4.1 E, J).

Some complex morphological structures (such as glia or muscles) were imported as surfaces and smoothed (Figure 4.1 C, H) using the *Smooth* modifier in Blender with a factor of three in all dimensions. Other structures (such as cilia or pigment cups) were manually approximated in Blender with geometrical shapes (Figure 4.1 B, G). All the imported objects were renamed, but the IDs from the TrakEM2 reconstructions were preserved as a part of the object name to assist with tracking back to the objects in the TrakEM2 project.

The detected synapses were exported from TrakEM2 in two files: one containing synapse positions and another containing information on the origin and target objects (pre- and postsynaptic cells). Synapses were represented in Blender as small spheres in their respective locations

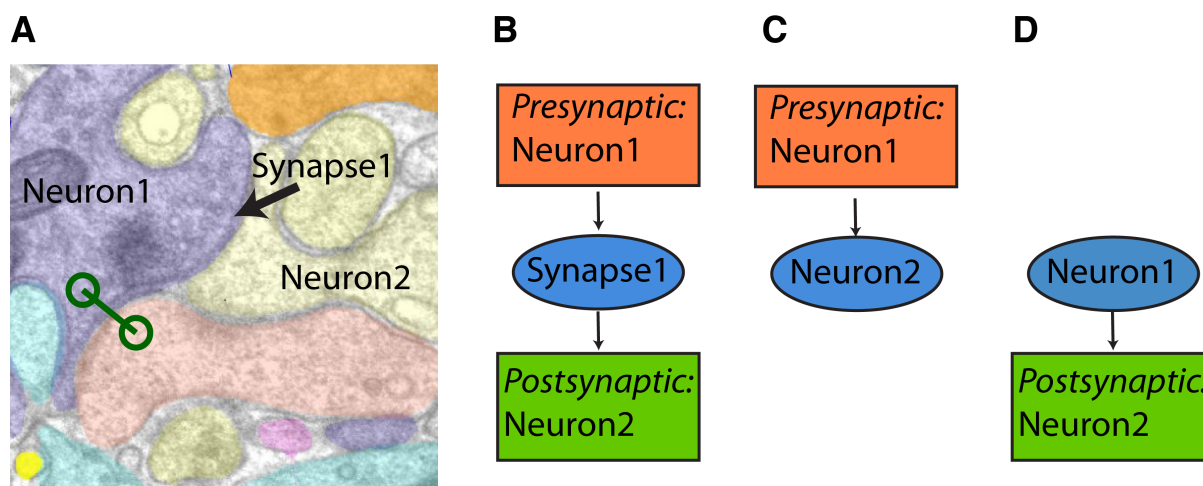


Figure 4.3.: Schematic importing of synaptic connectivity in a model.

Synapses detected in TrakEM2 (A) are imported as objects to Blender. Each synapse object has *Presynaptic* and *Postsynaptic* properties, where names of corresponding cell are stored (B). Subsequently, the Neuron1 is stored in the *Presynaptic* property of the Neuron2 (C) and the Neuron2 is stored in the *Postsynaptic* property of the Neuron1 (D).

(Figure 4.1 F, K). Synaptic connectivity information was stored using *ID-properties*. Blender *ID-property* is assigned to class instances, rather than to an entire class and remains available after restarting a project:

```
obj['Presynaptic'] = 'MNR2_159936|MNR1_159934|MN12_163184|MN11_159938'
```

Two ID-properties were created for each synapse: *Presynaptic* and *Postsynaptic*, containing lists of pre- and postsynaptic sites respectively (Figure 4.3 B). The imported cells were also extended with *Presynaptic* and *Postsynaptic* ID-properties containing respective lists of pre- and postsynaptic sites for each cell (Figure 4.3 C, D).

4.2.4. NeuroDetective functionality

Visibility property

The software stores and queries anatomical classification, behavior and gene expression information associated with reconstructed networks. The annotation was implemented using Blender grouping option, where objects of the same type or associated with the same behavior or gene were grouped together. Names of the groups denoting behavior start with underscore "_", names of the groups denoting expressed genes start with hyphen "-" and groups denoting anatomical classification start with any other character.

Blender native classes can be extended with custom properties of any type. Custom property is

not stored in the project file and needs to be redefined after restarting. Blender class *Group* was extended with a custom boolean property *Visibility*, which was used to implement showing and hiding groups of objects according to the user input (Figure 4.2 A-C):

```
BPY.TYPES.GROUP.VISIBILITY = BOOLPROPERTY(NAME='VISIBILITY', DEFAULT = TRUE)
```

An interface for setting visibility property for different groups was implemented using the Python API. Blender class *Panel* was extended to create an interface to show/hide the groups denoting anatomical classification, behavior and gene expression. Each panel contains a list of checkboxes for the corresponding groups (Figure 4.2 A-C). After user input, properties *hide* and *hide_render* are set to false for the objects checked as visible. For the remaining objects these fields are set to 'true'.

Network queries

Another panel was added to the NeuroDetective interface to query neuronal connectivity information. To display pre- or postsynaptic sites for a cell of interest, the information from the aforementioned *Postsynaptic* and *Presynaptic* ID-properties was used (Figure 4.2 F, 'Postsynaptic' and 'Presynaptic' buttons). The cells stored in the respective ID-property of a cell of interest are highlighted, whereas the remaining cells are hidden.

The software tool also queries for up- or downstream circuitry for a cell of interest (Figure 4.2 F, 'Post.subnetwork' and 'Pre.subnetwork' buttons). The neuronal network is considered as a directed graph, where neurons represent vertices and synapses represent directed edges. The depth-first search (DFS) graph traversal algorithm is then applied to find all up- or downstream cells (Algorithm 1). The retrieved cells are highlighted and the remaining cells are hidden.

NeuroDetective can highlight all in- or outgoing synapses for a cell of interest (Figure 4.2 F

Algorithm 1 *findSubnetwork()*

require: Connectivity graph G and current node v

return: Subnetwork for the node v

```

S.push(v)
visited.push(v)
while  $S$  is not empty do
     $u = S.pop()$ 
    if  $u$  not in  $visited$  then
         $visited.push(u)$ 
         $subnetwork.push(u)$ 
    end if
    for all  $w$  in  $G.adjacentNodes(u)$  do
         $S.push(w)$ 
    end for
end while
return  $subnetwork$ 

```

'Show in.connectors' and 'Show out.connectors' buttons). After user input, the synapses are highlighted, if they belong to the cell of interest. The remaining synapses are hidden.

Highlighting synapses between two selected cells was implemented (Figure 4.2 F 'Show connectors' button). After user input, synapses are highlighted if they have the first selected object as a presynaptic site and the second selected object as a postsynaptic site. The remaining synapses are hidden.

The application can highlight synapses between two groups (Figure 4.2 F 'Show connectors', Figure 4.5 D). After specifying presynaptic and postsynaptic groups of cells (Figure 4.2 F 'Pre.group' and 'Post.group' respectively) and pressing the button 'Show intergroup connectors', synapses are highlighted if their presynaptic and postsynaptic cells are in the respective groups. The remaining synapses are hidden. The selected cell groups are also highlighted.

Network centrality measures

The software tool can measure centrality for each neuron and highlight it using color-coding. Neuronal network was considered as a directed graph, where the neurons represent vertices and synapses represent directed edges. In- and out-degree centralities for a graph node were calculated as follows [176]:

$$CD(x) = deg(x),$$

where $deg(x)$ is the number of edges for a node x . All edges directed to the node were considered for in-degree centrality and all edges directed from the node were considered for out-degree centrality.

In- and out-closeness centralities were calculated for the nodes in a neuronal circuitry as follows:

$$CC(x) = \sum_{y \in G, y \neq x} \left(\frac{1}{d(x,y)} \right),$$

where $d(x,y)$ is a length of the path from the node x to the node y for out-closeness centrality and a length of the path from y to x for in-closeness centrality.

In- and out-eigenvector centralities were calculated as follows:

$$CE(x) = a * \sum_{y \in G, y \neq x} (g_{x,y} * CD(y)),$$

where a is a constant, $g_{x,y}$ is one, if nodes x and y are connected, and zero otherwise, $CD(y)$ is in- or out-degree centrality of the node y for in- and out-eigenvector centrality respectively.

Color-coding was generated as follows. The logarithm of centrality metric for each neuron was remapped to the range [0.3,1.0]. The range starting from 0.3 avoided excessively dark colors.

The RGB components of the color were then calculated as follows:

$$color = [0.6, 0.6 * (1 - lc), 0.6 * (1 - lc)],$$

where lc is the remapped logarithm of the centrality metric for a neuron. The resulting colors range from white (low centrality values) to bright red (high centrality values). Centrality was calculated using Python embedded in Blender. The Blender class *Panel* was extended to provide the interface (Figure 4.2 D).

Network statistics

NeuroDetective can assess statistical properties of the network elements, such as axon length, number of synapses or number of connections for a cell. Neuron length is determined as a sum of distances between the nodes in the longest branch of the skeleton. The number of outgoing or incoming synapses for a cell is determined as the number of synapses, for which that cell is specified as a presynaptic or postsynaptic site respectively. The number of pre- and postsynaptic sites for each cell in the network is defined as the number of unique elements in its *Presynaptic* or *Postsynaptic* ID-property respectively.

4.2.5. Mobile NeuroDetective

Mobile NeuroDetective was developed using the methods similar to those employed for the Mobile GeneExpressionAtlas (cf. Gene expression atlas for *Platynereis dumerilii*). The Android Developer Tools (ADT) Bundle was used, which includes Android Software Development Kit and a version of Eclipse Integrated Development Environment [106]. libGDX game development framework [107] was employed to visualize the 3D model on the Android platform. The number of vertices in the Blender model was reduced by applying the *Decimate* modifier to each object. Ciliary band cells in the outer circle were also removed to optimize the model. The Blender model was exported in OBJ format, converted in GD3D format using the FBX converter [108], and then loaded in Mobile NeuroDetective using libGDX. Mobile NeuroDetective class diagram is identical to the one described in Gene expression atlas for *Platynereis dumerilii*. The application ran successfully on the Samsung Galaxy Note 10.1 tablet and Samsung Galaxy S III Mini.

4.2.6. Online NeuroDetective

The web application Online NeuroDetective was developed using the methods similar to those employed for the Online GeneExpressionAtlas. HTML and JavaScript [109] programming languages and the Three.js library [110] were employed. First, the neuronal circuitry model was optimized for the web application by applying the *Decimate* modifier to each object. It was then exported from Blender in .js (javascript) format and then loaded in the application using the Three.js library, which enabled an interactive 3D visualization. The application was extended with an option to manipulate the model view using the OrbitControls library in Three.js. Setting the visibility option for the anatomical groups was implemented (Figure 4.14 B). Exploring a neuronal network model using Online NeuroDetective requires WebGL library [112] enabled in a web browser.

4.2.7. Analysis of eye-ablation experiments

The eyes were ablated on one side of the *Platynereis* larvae. The ablation was performed on an Olympus FV1000 confocal microscope equipped with a 351-nm pulsed laser (Teem Photonics). After 1-6h of recovery in seawater, the larvae were placed in a cuvette with seawater for the phototaxis experiments. The cuvette with the larvae was illuminated from one side. Swimming behavior was recorded using a Zeiss stereomicroscope. Non-ablated larvae from the same batch were used as a control. Nadine Randel and Gáspár Jékely performed eye-ablation experiments.

Larval response was characterized by calculating the number of closed loops in the swimming trajectories throughout the video recording. For that, the background was determined by taking a median of the corresponding pixels in the video frames. The background was then subtracted from each frame (Figure 4.4 A, B). The larvae were segmented from the processed video using the MaxEntropy threshold method in Fiji and subsequently tracked using the MTrack2 [177] plugin. The number of closed loops was calculated as a number of trajectory intersections. The loop was counted when the larva crossed a part of its trajectory. The part of the trajectory that had been involved in a previous loop was not used for detecting the subsequent intersections to avoid counting false intersections of the overlapping loops. For example, the trajectory part marked in red in the Figure 4.4 E is not considered for the subsequent loop detection and its intersection with the blue part of the trajectory is not counted as a loop (Figure 4.4 E). A threshold for the minimum loop radius was set to eliminate larvae remaining in one position.

The metric was calculated as follows:

$$PtT = \frac{loopCount}{trajectoryCount}$$

The algorithm was implemented as a plugin for Fiji [101] using java programming language [178].

4.2.8. Analysis of light-contrast experiments

Platynereis larvae at 72 hpf were placed in a 6x15x3 mm rectangular glass cuvette filled with seawater. The cuvette was illuminated with two light sources of different intensity from opposite sides. The brighter light source came from the right side, relative to the camera. Swimming behavior under different light contrasts was recorded using a Zeiss stereomicroscope. Light-contrast experiments were performed by Nadine Randel and Gáspár Jékely.

To characterize the effect of light contrast on *Platynereis* phototaxis, a custom Fiji plugin was developed. Phototaxis magnitude was measured as an average displacement of larvae along the

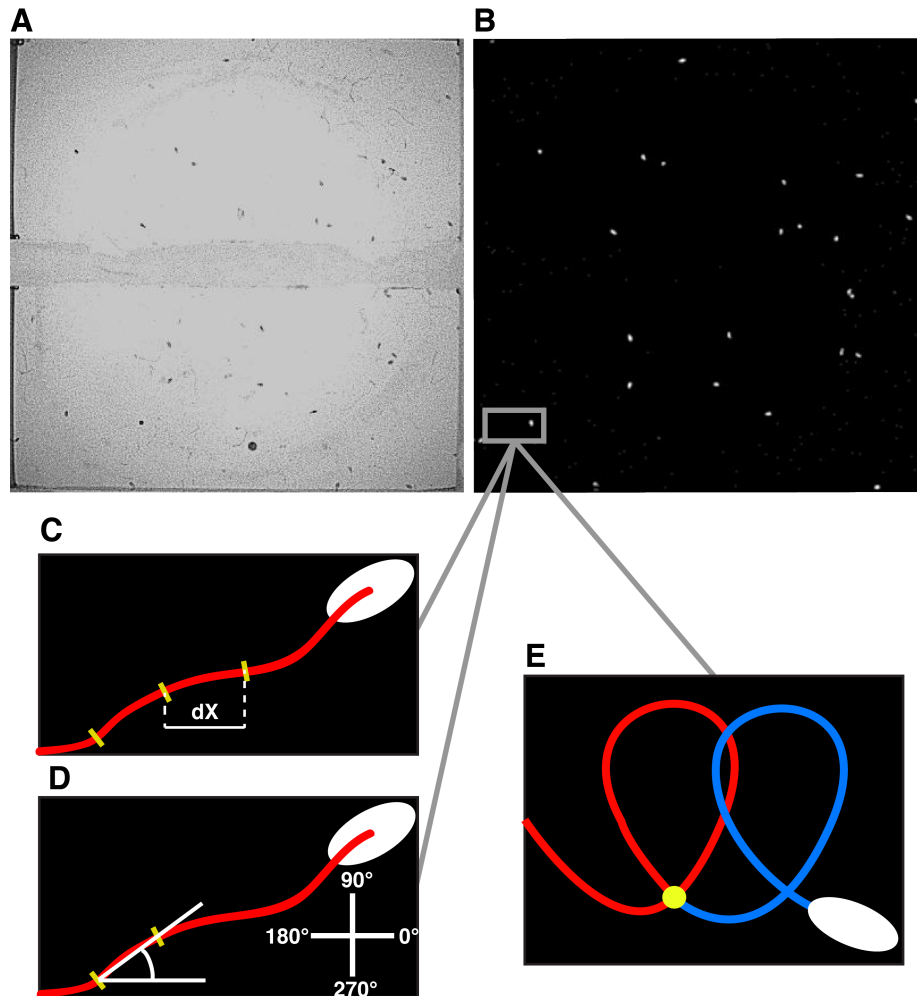


Figure 4.4.: Analysis of the phototaxis experiments.

(A) An individual frame from a video recording of a phototaxis experiment. (B) A video frame with the background removed. (C) Calculation of the displacement for an individual trajectory fragment. (D) Calculation of the direction for an individual trajectory fragment. (E) Counting closed loops in the individual larva trajectory. Trajectory intersection is considered as a closed loop, overlapping loops are not counted.

illumination axis. The background was determined by taking a median of the corresponding pixels in all the video frames and subsequently subtracted from each frame (Figure 4.4 A, B). The larvae were then segmented using the MaxEntropy threshold method in Fiji and tracked using the MTrack2 plugin. Each determined trajectory was broken down into fragments of a minimum length of 4 pixels ($\sim 250 \mu\text{m}$), which is comparable to larva size in the video recordings. Displacement along the illumination axis was calculated for each trajectory fragment (Figure 4.4 C). Final phototaxis metric was calculated as:

$$PtM = \frac{dx^+ - dx^-}{dx^+ + dx^-},$$

where dx^+ is the sum of positive displacements along the illumination axis for all larva trajectories and dx^- is the sum of negative displacements. Overall dynamics of swimming direction throughout the experiment was measured as distribution of the fragment

direction was calculated as an angle between the trajectory displacement vector and the illumination axis (Figure 4.4 D).

4.3. Results

In the frame of my PhD project I developed a software tool named NeuroDetective that analyzes neuronal networks and extracts information from the large volumes of complex electron microscopy data in an efficient and convenient way. The software largely automatizes the process of modeling reconstructed neuronal networks and provides an efficient way to visualize neuronal morphology and spatial distribution of the elements in a neuronal circuitry. It incorporates synaptic connectivity information and provides a coherent way to access various aspects of connectivity information. NeuroDetective extends a model with attributes associated with a neuronal network and provides this information in the course of network exploration.

NeuroDetective was applied to the reconstructed neuronal circuitry of a 72 hours post-fertilization (hpf) *Platynereis* specimen in order to explore the neuronal mechanisms involved in larval phototaxis or larval response to mechanical stimuli. Analyzing connectivity of the reconstructed network revealed cells, connections and motifs important for the network activity. Behavioral experiments were then analyzed to verify the hypothetical neuronal mechanisms derived from the neuronal network exploration. The model was provided for public access via a web application, an Android device application and the Blender model file.

4.3.1. NeuroDetective: a software tool for exploring neuronal connectivity

NeuroDetective models the neuronal network elements and represents their morphology in a simplified manner, thereby compensating for the imperfections in the alignment and providing a smooth surface representation (Figure 4.1). The model can then be extended with neuronal connectivity information and annotated with anatomical classification, behavioral and gene expression information. The software allows conveniently accessing and querying all available information, thereby providing an extensive tool for the exploration of neuronal circuitry.

Generating a 3D model of a neuronal network

NeuroDetective automates the process of developing a 3D model for a neuronal circuit. It automatically imports neuron reconstructions in OBJ, a common format for representing 3D shapes (cf. Neuronal network model, Figure 4.2 E). The algorithm represents the cell bodies

of reconstructed neurons as spheres and the axons and the dendrites as tubes (Figure 4.1 E, J). The skeletons of the axons and dendrites are smoothed using an average filter to overcome imperfections in the alignment of electron microscopy sections. Anatomical structures of a complex shape can be loaded into a model as surfaces using native Blender functionality or can be manually approximated with geometric shapes (Figure 4.1 B, C, G, H). In the latter case, biological structures can be provided in any geometry-definition format supported by Blender (e.g. OBJ, 3DS and STL).

Connectivity information is crucial for analyzing the flow of electrochemical activity within a neuronal network and understanding its functionality. Therefore, I extended NeuroDetective with the option to automatically import, store and query neuronal connectivity information (cf. Neuronal network model, Figure 4.2 E). The software loads synaptic connectivity information from text files exported from TrakEM2 and represents individual synapses as spheres in their corresponding positions in 3D space (Figure 4.1 F, K). Synaptic connectivity is integrated in the model by annotating each cell with information about its pre- and postsynaptic sites. The spheres representing synapses are also annotated with information about their pre- and postsynaptic sites (cf. Neuronal network model).

Extending a model with additional information

NeuroDetective can be adjusted to a specific project, for example, it can be extended with additional information important for the analysis of a particular neuronal network. Each element of the network model can be annotated with information of a certain type. I implemented an interface to set the visibility option for the groups in a given classification (Figure 4.2 A-C). For the purposes of the current project, NeuroDetective was extended with an option to annotate a model with anatomical classification and information on gene expression and organism behavior. The software is not restricted to these classifications and can be further extended according to the needs of individual projects.

Exploring neuronal circuitry

Visualizing synapses reveals their position relative to the cell body of a postsynaptic cell, which can elucidate its role as an excitatory or inhibitory synapse [179]. Therefore, I implemented an option to highlight synapses for cells of interest. NeuroDetective can query all incoming or outgoing synapses for a cell of interest (Figure 4.5 A), reveal all synapses between two cells (Figure 4.5 B) or display all the synapses between two cell groups (Figure 4.5 C, D, cf. NeuroDetective functionality).

To facilitate exploring information flow within complex neuronal networks, NeuroDetective includes an option to query various aspects of synaptic connectivity (Figure 4.2 E, cf. NeuroDetective functionality), such as retrieving pre- and postsynaptic sites directly connected to a cell of interest (Figure 4.5 E), or highlighting the complete down- and upstream circuitry of a cell (Figure 4.5 F).

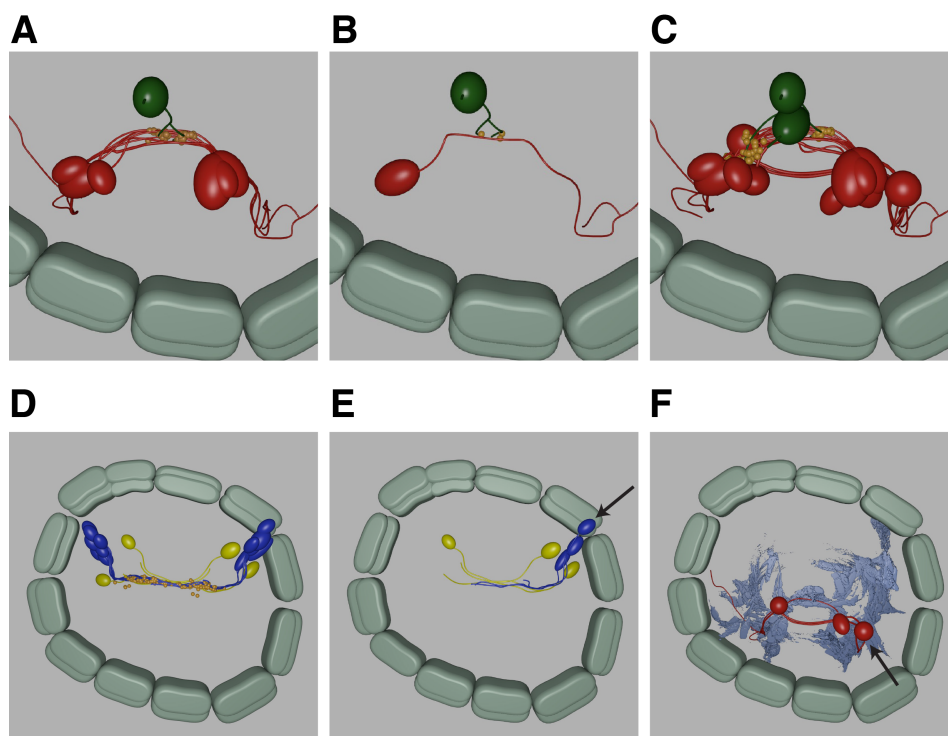


Figure 4.5.: NeuroDetective functionality for exploring synaptic connectivity. (A) All outgoing synapses (orange) and the postsynaptic motoneurons (red) of a mechanosensory cell (green, marked with the arrow). (B) Synapses (orange) from the mechanosensory cell (green) to the motoneuron (red). (C) All synapses (orange) from the mechanosensory cells (green) to the motoneurons (red). (D) All synapses (orange) from the eye photoreceptors (blue) to the primary interneurons (yellow). (E) All postsynaptic sites of the eye photoreceptor cell (marked with the arrow). (F) All cells downstream of the motoneuron (marked with an arrow). Color-code as in Figure 4.9.

Statistical characterization of the neuronal circuitry.

NeuroDetective was extended with tools for statistical analysis of neuronal networks using the Python API enabled in Blender. The software calculates various statistical properties, such as distribution of incoming and outgoing synapses and pre- and postsynaptic sites for all cells or cells in a specific group. It also calculates network centrality for individual neurons and color-codes them according to the metric. This option maps network centrality in the context of the neuronal network anatomy.

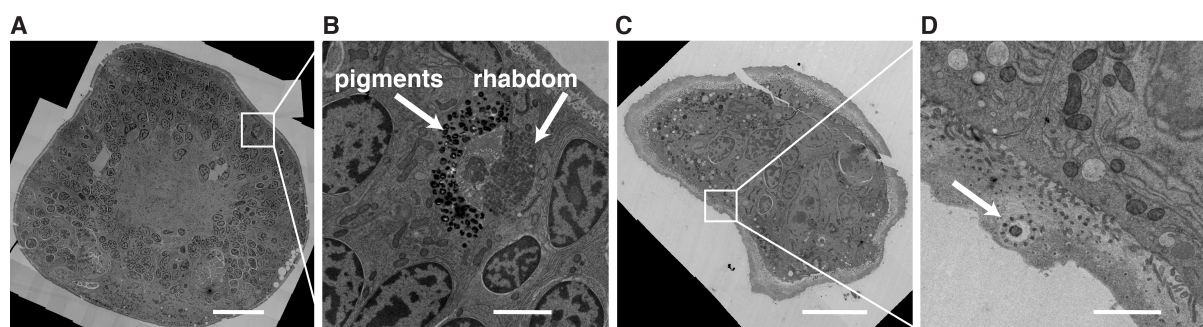


Figure 4.6.: Adult eyes and mechanosensory cells in the electron microscopy images. (A, C) Individual layers from the electron microscopy image stack at the level of the adult eyes (A) and the mechanosensory cells (C). (B, D) Close-up images of the eye photoreceptors and pigment cells (B) and a mechanosensory cell (D) (marked with an arrow). Scale bar 20 μm in A and C, 2 μm in B, 1 μm in D.

4.3.2. Model for *Platynereis* neuronal network

The neuronal circuitry downstream of the photoreceptor cells (Figure 4.6 A, B, 4.7 A) and downstream of the mechanosensory cells in the *Platynereis* apical organ (Figure 4.6 C, D, Figure 4.7 B) was reconstructed from the electron microscopy images of a 72 hpf *Platynereis* specimen. The photoreceptors and the mechanosensory cells exhibit distinctive morphology and could be visually detected in the electron microscopy images (Figure 4.6). Analyzing these circuits reveals neuronal mechanisms regulating behavioral response to visual and mechanosensory stimuli in *Platynereis*. Additionally, ciliary photoreceptors (Figure 4.7 C), larval eye photoreceptors (Figure 4.7 D) and cells expressing myoinhibitory peptide (MIP) (Figure 4.7 E) were reconstructed.

A 3D model of the reconstructed neuronal circuitry was created in Blender using functionality provided by NeuroDetective. Using automatic import and hand-drawing (cf. Neuronal network model), 94 neurons from the reconstructed neuronal network were represented in the 3D model (Table 4.2 and Figure 4.1 D, E, I, J). Some anatomical parts with a complex shape (such as muscles and glia) were loaded into the model as surfaces in OBJ format and smoothed (Figure 4.1 C, H). Other anatomical parts (such as cilia or pigment cups) were manually approximated with geometric shapes (Figure 4.1 B, G). These cells provided anatomical landmarks for neuronal network exploration.

1143 synapses were imported into the model of the reconstructed circuit (Figure 4.1 F, K). Visualizing synapses revealed anatomical regions with higher information exchange rates between neurons, which are highly important for network functionality. The spatial distribution of synapses in the *Platynereis* visual circuitry outlines primary and secondary optic neuropils, where most information exchange takes place (Figure 4.1 K). The spatial distribution of the synapses also provides additional anatomical landmarks.

4.3.3. Analysis of synaptic connectivity in *Platynereis*

Information flow in the reconstructed network was analyzed using the tools provided in NeuroDetective. The signal from the photoreceptor cells (Figures 4.9 and 4.8) passes via the primary interneurons (primary IN) to the trans-optic-neuropil interneurons (transoptic IN) (the interneurons that project into a secondary optic neuropil). The transoptic interneurons connect to the Schnoerkel interneurons (Schnoerkel IN), which in turn synapse onto motoneurons. Finally, the motoneurons synapse on the longitudinal trunk muscles. There are other connections in the network that modulate signal propagation, for example, the dorsal interneurons (dorsal IN) that project onto the motoneurons.

The apical mechanosensory cells connect directly to the motoneurons from the visual circuitry, which synapse on the longitudinal trunk muscles (Figures 4.7 B and 4.8). Ciliary photoreceptors propagate the signal to the muscles via general interneurons, Schnoerkel interneurons and motoneurons (Figures 4.7 C and 4.8). The larval eyes show strong connection to the motoneurons, which, as previously mentioned, connect to the muscles (Figures 4.7 D and 4.8).

I analyzed the distribution of the incoming and outgoing synapses for the cells in the neuronal network. The average number of outgoing synapses per cell is 14.9 with a maximum of 116 synapses for an individual primary interneuron (Figure 4.10 B), whereas the average number of incoming synapses per cell is 15.6 with a maximum of 64 (an individual motoneuron) (Figure 4.10 B).

There are 13 cells without any presynaptic cells in the reconstructed network, most of which

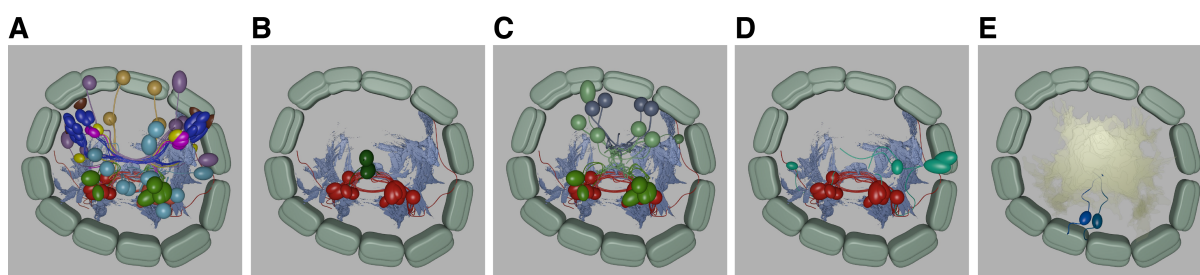


Figure 4.7.: Elements of *Platynereis* neuronal circuitry reconstructed from the electron microscopy images.

(A) The reconstructed visual circuitry (detailed description in Figure 4.9). (B) The circuit downstream of the mechanosensory cells (green), propagating signal to the motoneurons (red) and subsequently the muscles (light blue). (C) The circuit downstream of the ciliary photoreceptors, propagating signal to the ventral interneurons (light green), the Schnoerkel interneurons (green), the motoneurons (red) and finally the muscles (light blue). (D) The circuit downstream of the larval eyes, propagating signal to the motoneurons (red) and the muscles (light blue). (E) The cells expressing myoinhibitory peptide (MIP) (blue) with the reconstructed plexus as an anatomical landmark.

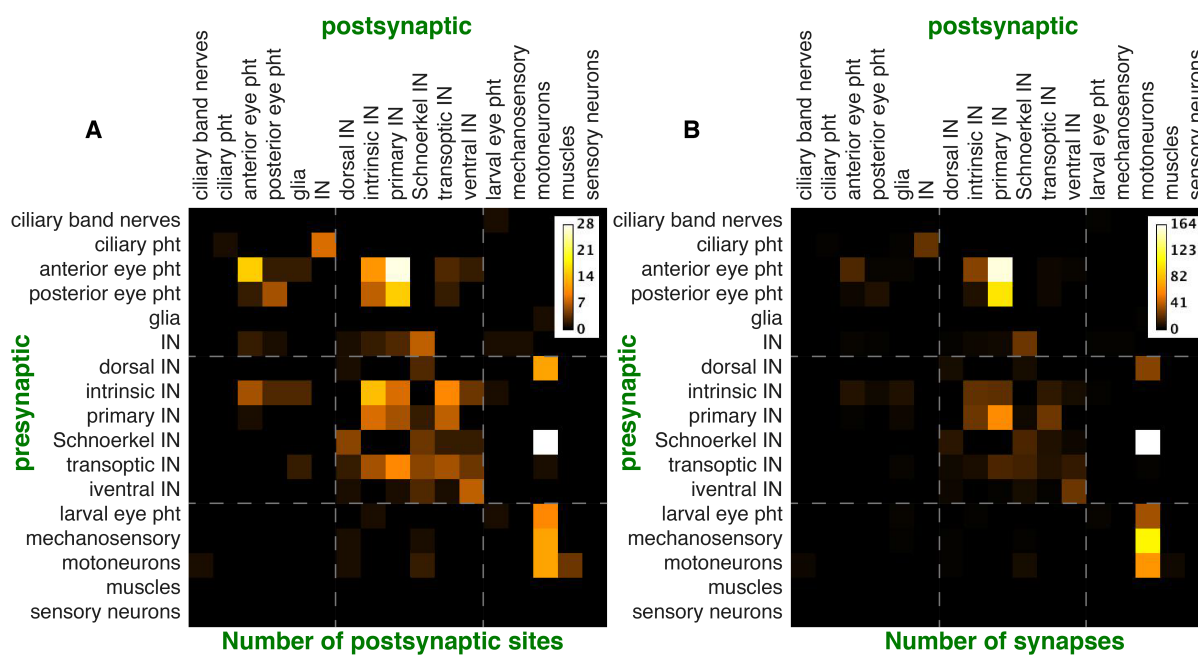


Figure 4.8.: Connectivity matrices for the anatomical groups.

(A) Number of cell-cell connections between anatomical groups. (B) Number of synapses between cells from different anatomical groups. IN, interneurons; pht, photoreceptors.

are sensory cells (photoreceptors and mechanosensory cells). There are 15 terminal cells (i.e. without postsynaptic cells in the reconstructed network), among which there are some interneurons (whose postsynaptic sites might not yet be reconstructed) and muscles.

Next, I analyzed the number of pre- and postsynaptic sites for each cell and for each group. The average numbers of both pre- and postsynaptic sites for a cell are distributed around 3.6 with typical values in the range of 0 to 10 (Figure 4.10 A, B). On average, the mechanosensory cells and the larval and adult eye photoreceptors show low numbers of incoming synapses and high numbers of outgoing synapses (Figure 4.11 A). This is likely due to their sensory role in the neuronal network.

On average, the primary interneurons exhibit prominently high numbers of pre- and postsynaptic sites, which in conjunction with high numbers of incoming and outgoing synapses (Figure 4.11) demonstrates vast and strong connections of these neurons in the circuitry and implies their importance in the visual circuitry. The ventral interneurons show low numbers of pre- and postsynaptic sites and incoming and outgoing connections, which suggests a weak influence on the network functionality (Figure 4.11). The average number of input and output synapses are similar for most groups of interneurons, which reflects their transmissive role in the network (Figure 4.11 A). No significant correlation ($r=0.22$) was determined between the number of pre- and postsynaptic sites.

I measured the axonal length (cf. NeuroDetective functionality) of the photoreceptor cells,

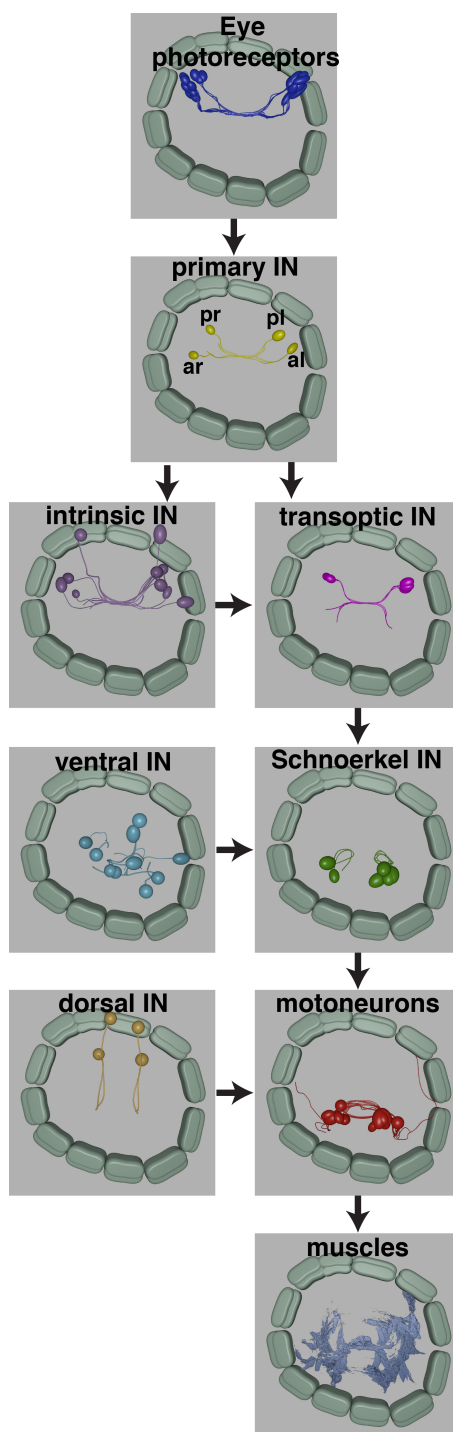


Figure 4.9.: Information flow within the visual circuitry in *Platynereis* larva.

The photoreceptor cells pass the signal via the primary interneurons (primary IN) to the transoptic interneurons (transoptic IN). The latter connect to the Schnoerkel interneurons (Schnoerkel IN), which in turn synapse onto motoneurons. Finally, the motoneurons provide signal for the longitudinal trunk muscles. The signal propagation is modulated by the intrinsic interneurons (intrinsic IN) projecting on the transoptic interneurons and the dorsal interneurons (dorsal IN) projecting on the motoneurons. IN, interneurons; pht, photoreceptors; pr, posterior right; pl, posterior left; ar, anterior right; al, anterior left.

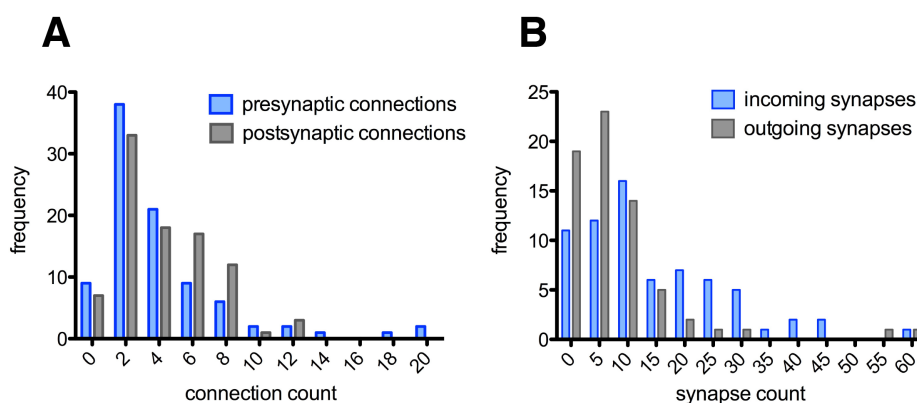


Figure 4.10.: Distribution of neurons by the number of connections and synapses. (A) Distribution of neurons by the number of pre- and postsynaptic connections. (B) Distribution of neurons by the number of incoming and outgoing synapses.

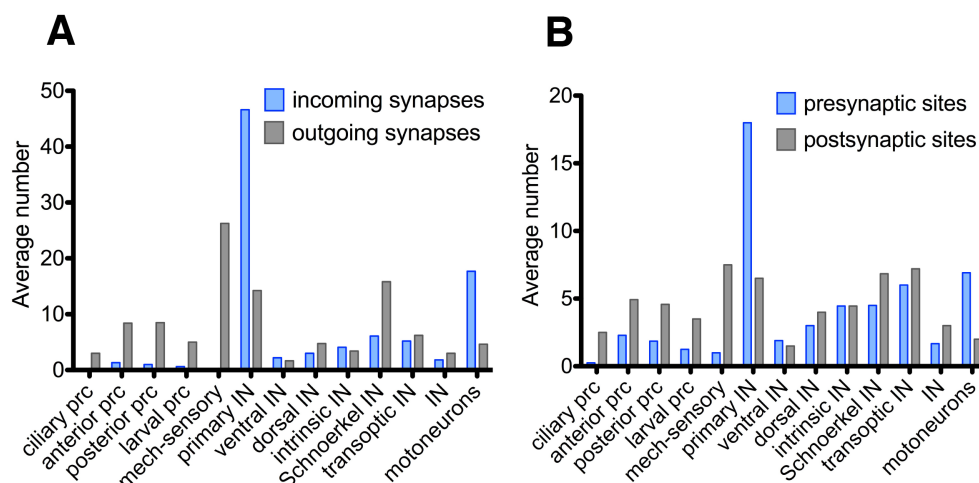


Figure 4.11.: Distribution of connections and synapses by anatomical groups. (A) Distribution of the average number of pre- and postsynaptic connections by anatomical groups. (B) Distribution of the average number of incoming and outgoing synapses by anatomical groups. IN, interneurons; pht, photoreceptors.

which showed positive correlation with rhabdom volume ($r=0.63$, $p=0.002$) [12]. The correlation suggested a more mature stage of development for cells with longer axons. This feature characterizes the development of the synaptic connectivity in the course of neuron growth [12].

4.3.4. Centrality measure for *Platynereis* neuronal network

Degree, eigenvector and closeness centralities were measured for the elements of the *Platynereis* neuronal circuit (Figure 4.12). In- and out-degree centralities were calculated as the number of incoming and outgoing synapses respectively. High out-degree centrality was demonstrated by the sensory cells (the mechanosensory cells and the eye photoreceptors) and by the Schnoerkel

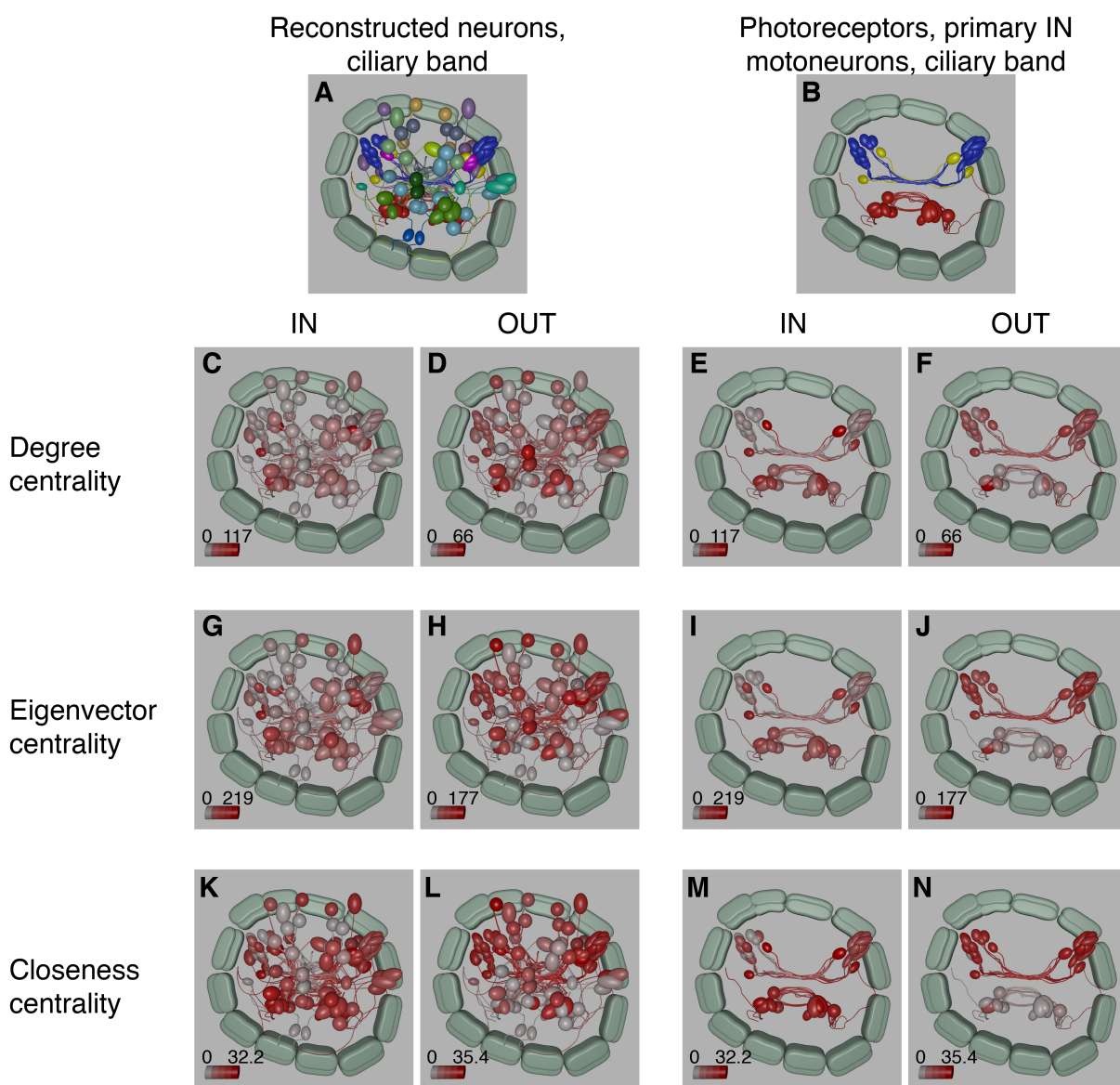


Figure 4.12.: Centrality measure for *Platynereis* neuronal network

(A) The complete model of the reconstructed network color-coded according to the anatomical classification. (B) The eye photoreceptors (blue), the primary interneurons (yellow) and the motoneurons (red) together with the ciliary band (gray). (C, E) In-degree centrality. (D, F) Out-degree centrality. (G, I) In-eigenvector centrality. (H, J) Out-eigenvector centrality. (K, M) In-closeness centrality. (L, N) Out-closeness centrality. (C, D, G, H, K, L) The complete reconstructed network color-coded according to a centrality metric. (E, F, I, J, M, N) The eye photoreceptors, the primary interneurons and the motoneurons color-coded according to a centrality metric.

interneurons (Figure 4.12 B, D). The in-degree value was rather homogenous for the neuronal network (Figure 4.12 A, C), with the exception for the primary interneurons, which receive strong input from the eye photoreceptors, and some motoneurons, which receive strong input from mechanosensory cells (Figure 4.8).

Eigenvector centrality quantifies the influence of a node in a network and reflects its connectedness to high-degree nodes. In-eigenvector centrality was low for the sensory cells, which do not receive much input from highly connected neurons (Figure 4.12 E, G). The interneurons (primary, transoptic and Schnoerkel) and the motoneurons receive input from highly connected neurons and, thus, showed higher level of in-eigenvector centrality. The network elements largely exhibited high out-eigenvector centrality (Figure 4.12 D, F), which demonstrated high network connectedness. The motoneurons represented one of the exceptions (Figure 4.12 F), having most synapses to the muscles that did not synapse to other elements in the reconstructed network.

Closeness centrality reflects the ability of a node to spread information to other elements in a network. It is defined as a sum of inverse distances to other nodes. The sensory elements of the network revealed low in-closeness and high out-closeness centrality (Figure 4.12 H-K), which showed that they can spread information to many cells, but not many cells can reach them. Conversely, the motoneurons cannot pass information to many cells, but receive input from many cells, which was reflected in their high in- and low out-closeness centrality (Figure 4.12 H-K). The interneurons, especially the primary interneurons, exhibited high values for in- and out-closeness centrality (Figure 4.12 H-K), which suggested their strong influence over other elements in the network.

4.3.5. Additional information for the *Platynereis* neuronal network

The generated model of *Platynereis* neuronal circuitry was extended with the anatomical classification and the information on behavior and gene expression. The anatomical classification was established in ref. [12] based on anatomical and physiological properties (Figure 4.2 A), which resulted in 20 cell types (Table 4.2). Annotating the model with the anatomical classification [12] allowed the analysis of the spatial distribution of cells of different types. Most of the cell types exhibited high bilateral symmetry (for example, eye photoreceptors, primary, transoptic and dorsal interneurons and others). However, several neurons showed asymmetrical spatial distribution (for example, ventral and intrinsic interneurons).

Information on the expressed genes was obtained from *in situ* hybridization specimens and subsequent matching of the obtained signal to the electron microscopy data based on cell morphology and anatomical landmarks. The expression of rhabdomeric opsin1 (r-opsin-1) was

Cell type	Cell count
ciliary band	24
ciliary band nerves	1
ciliary photoreceptors	4
connectors	1143
anterior eye photoreceptors	14
posterior eye photoreceptors	7
glia	3
interneurons	6
dorsal interneurons	4
ventral interneurons	10
intrinsic interneurons	11
primary interneurons	4
Schnoerkel interneurons	6
transoptic interneurons	5
larval eye photoreceptors	4
mechanosensory neurons	2
motoneurons	11
muscles	-
pigment cup	4
sensory neurons	2

Table 4.2.: Anatomical classification of the elements in the *Platynereis* neuronal circuitry.

determined to be in the adult eyes based on *in situ* hybridization and assigned to the eye photoreceptors in the neuronal circuitry model, based on the specific morphology of these cells in the electron microscopy images. Similarly, MIP- and RGW- expressing cells were determined by *in situ* hybridization and then matched to their corresponding cells in the electron microscopy images based on morphology and relative position to anatomical landmarks detectable in both light and electron microscopy.

Identification of gene coexpression also allows the assignment of gene expressions to their corresponding cells in the neuronal network model. For example, the *TRP-CP* gene was shown to coexpress with *r-opsin-1* based on light microscopy data (cf. Gene expression atlas for *Platynereis dumerilii*), therefore this gene was also assigned to the eye photoreceptors in the model. Another approach is to register gene expression patterns obtained from *in situ* hybridization and light microscopy to the electron microscopy data using cell nuclei as a reference signal (cf. Integrating light and electron microscopy datasets). Expression patterns of seven genes were determined and introduced into the model.

4.3.6. Mobile and web applications for the *Platynereis* neuronal network

The model of the reconstructed *Platynereis* nervous system was provided as web and Android applications, called NeuroDetective (Figure 4.14) and Mobile NeuroDetective (Figure 4.13) respectively. Both applications can query across anatomical classification (Figure 4.13 C, Figure 4.14 B). The Blender model was optimized by applying the *Decimate* modifier to each object, which reduced the number of vertices in the objects without introducing visual artifacts. It sped up the applications on devices with limited computational resources and improved user experience. The Three.js library [110] was employed to visualize the 3D model in the web browser and the libGDX [107] library was used for visualization on Android devices. The applications efficiently access the neuronal reconstruction data without a need to install Blender and master its interface. The applications provide a brief overview of the neuronal circuitry, which can be further explored by downloading the Blender model.

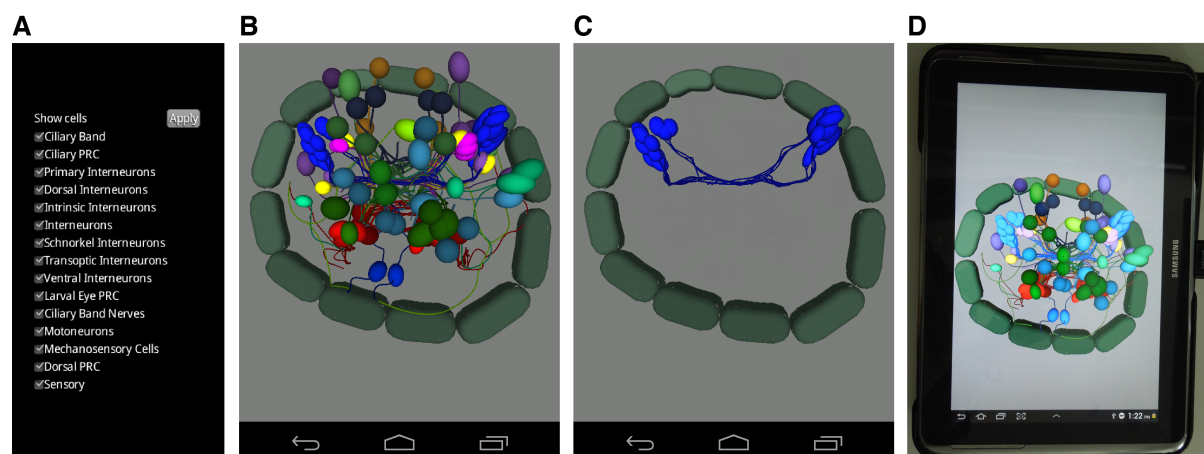


Figure 4.13.: Mobile NeuroDetective.

(A) The main menu screen for selecting the groups of objects for visualization according to the anatomical classification. (B, C) A screen for the interactive 3D visualization of the objects selected in the main menu. (B) All objects in the model of the reconstructed neuronal network. (C) The eye photoreceptors (blue) together with the ciliary band (gray) selected for visualization. (D) A photograph of a tablet running the mobile NeuroDetective.

4.3.7. Neuronal mechanisms of phototaxis in *Platynereis*

Defining sensory elements in the visual circuitry

Behavioral experiments were performed to verify whether dorsal eyes and their downstream neuronal circuit modulate phototaxis in *Platynereis*. The eyes were ablated on one side of the *Platynereis* larvae, which were then placed in a cuvette (cf. Analysis of eye-ablation exper-

4. Neuronal circuitry model for *Platynereis dumerilii*

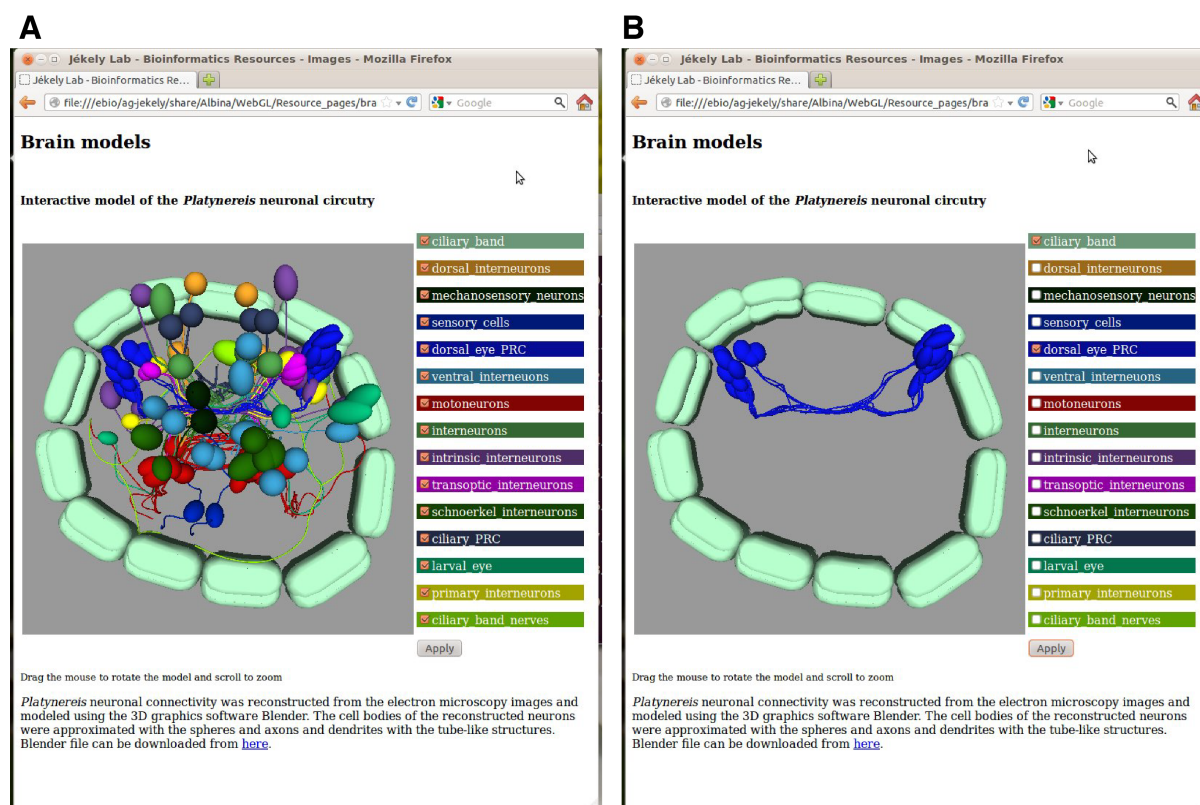


Figure 4.14.: Online NeuroDetective, a web application for exploring the neuronal network. Screenshots of Online NeuroDetective running in the web browser, which presents an interactive 3D view of a neuronal circuitry model and a list of anatomical groups that can be selected for visualization. (A) All objects in the neuronal network model selected for visualization. (C) The eye photoreceptors (blue) with the ciliary band (gray) selected for visualization.

iments). The cuvette was illuminated from one side and the swimming response was video-recorded. Control *Platynereis* larvae were positively or negatively phototactic and followed a straight trajectory, whereas ablated larvae exhibited a spiral-like swimming path.

I developed a Fiji plugin to automatically characterize swimming behavior in eye-ablated *Platynereis* larvae. The metric was based on the number of closed loops performed by swimming larvae. Visual inspection demonstrated that the metric reflected the swimming dynamics in the video recordings. The video recordings with the larvae following a spiral-like trajectory showed a 10-fold higher metric than the video-recordings with the larvae following a straighter trajectory (Figure 4.15 A, B).

The plugin was applied to the video recordings of 8 experiments with control and eye-ablated larvae. The resulting metric values for ablated larvae were significantly higher than for control larvae (Figure 4.15 C, $p = 0.0003$). The plugin demonstrated, that ablation of eye photoreceptors on one side changes swimming behavior of the larvae, as they are no longer able to follow the light vector. These results confirmed that the neuronal circuitry downstream of the adult eyes modulates phototactic behavior in *Platynereis* at the early developmental stages and that

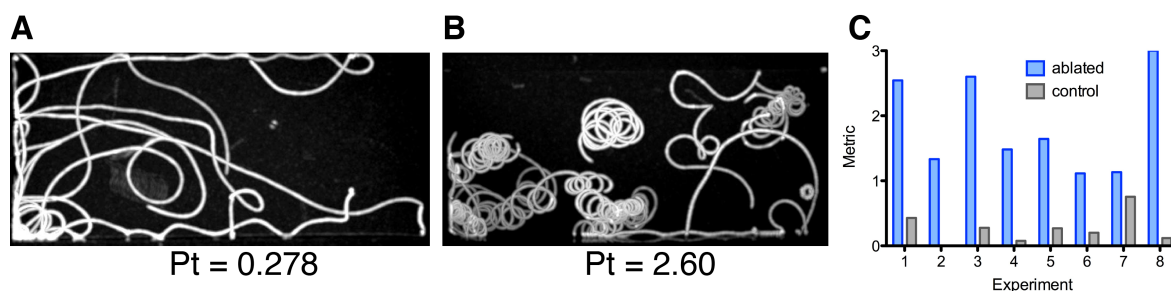


Figure 4.15.: Eye-ablation experiment analysis.

(A) The maximum-intensity projection of the phototaxis-experiment video with control larvae. The background is removed. The phototaxis measure for the video is 0.278. (B) The maximum-intensity projection of the phototaxis-experiment video with eye-ablated larvae. The background is removed. The phototaxis measure for the video is 2.6. (C) Phototaxis metric for eight experiments involving control and eye-ablated larvae.

the adult eyes on both sides are necessary to generate an accurate behavioral response.

The obtained behavioral information was integrated into the Blender model. This information can be accessed in the course of exploration of the neuronal circuit via the interface provided in NeuroDetective.

Information flow in the visual circuitry

Reconstructing the visual circuitry in *Platynereis* revealed a contralateral reciprocal communication between the primary interneurons (Figure 4.9), where anterior right and posterior left interneurons synapse on each other and anterior left and posterior right interneurons synapse on each other [12]. Furthermore, the synapses lie closer to the cell body of the postsynaptic neuron, suggesting an inhibitory nature of the synapses [179]. Such architecture suggests that *Platynereis* larvae respond to light contrast rather than to absolute light intensity. If larvae detect light of the same intensity coming from different directions, the cross-wise inhibition suppresses the signal and hinders its propagation to the muscles. In contrast, if the light coming from one direction has lower intensity, it can not compensate for the activation on the opposite side and the signal reaches the muscles, thereby causing body bending and a subsequent trajectory change. To verify this hypothesis, behavioral experiments were performed (cf. Analysis of light-contrast experiments) where two light sources of different intensities were applied from the different sides of a cuvette with freely-swimming larvae. Multiple light contrast levels were applied and the respective behavioral response was recorded.

To facilitate the analysis of the behavioral experiments and increase its scale, I developed a plugin to measure the effect on the larval swimming behavior of changing contrast levels. The plugin characterizes phototactic response based on average displacement of the larvae along the

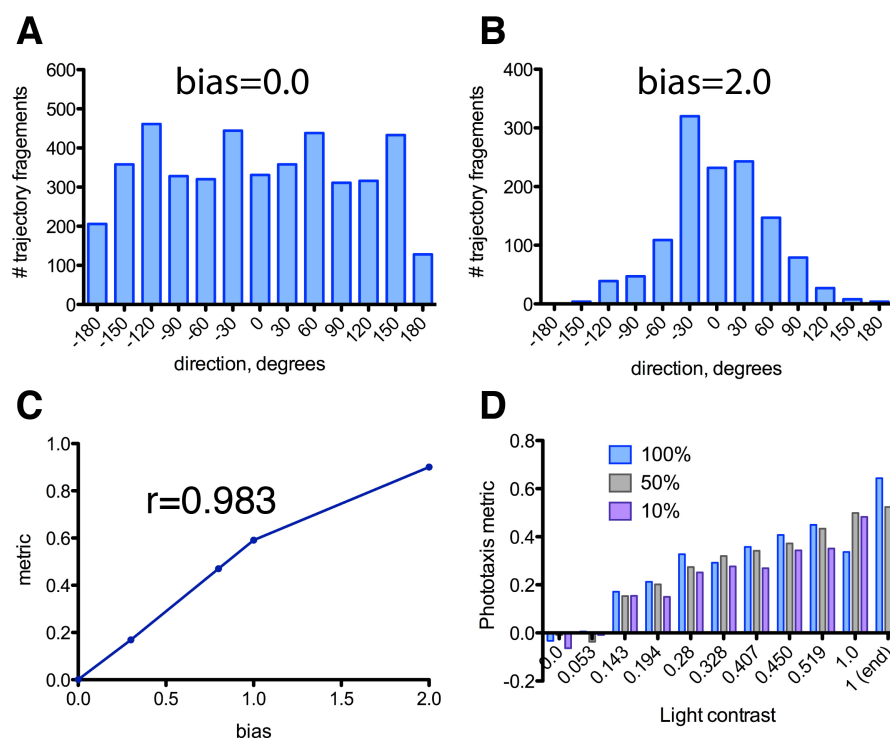


Figure 4.16.: Light-contrast phototaxis experiment.

(A, B) The distribution of the trajectory fragments by swimming directions in the simulation videos. The bias values ranged from 0.0 (A) to 2.0 (B). (C) The phototaxis metric for the generated videos with simulated larvae with the bias values ranging from 0.0 to 2.0. (D) The phototaxis metric for the experiments with the contrast levels ranging from 0.0 to 1.0 and varying initial light intensity (10% (purple), 50% (gray) and 100% (blue)).

illumination axis (cf. Analysis of light-contrast experiments). The average displacement was calculated over trajectory fragments, rather than entire trajectories. Such metric reflects the fact that larvae can change their swimming direction several times in the course of the experiment. I evaluated the method on simulated experimental videos, where 50 ovals imitating swimming larvae were moving around throughout 500 frames. The movement of the artificial "larvae" was generated randomly in the range of 5 pixels along x and y direction with a) a bias of zero pixels towards one side along the X-axis, b) with a bias of 0.3 pixels, c) with a bias of 0.8 pixels, d) with a bias of 1 pixel, e) with a bias of 2 pixels (cf. Figure 4.16 C). The results showed that the metric correlates with the introduced displacement bias and reflects the change in the swimming dynamics (Pearson $r=0.983$, $p=0.0002$).

I also defined the swimming-direction distribution for the trajectories of the simulated "larvae", which exhibited a uniform distribution of angles for the random displacement simulations with a bias of zero pixels (Figure 4.16 A) and a bell-shape distribution centered around 0 degrees for other simulations (direction to the right side of the video). The distribution becomes taller and narrower with the increase of shift bias (Figure 4.16 B).

I applied the developed plugin to the video recordings of the phototaxis experiments and dis-

covered that the magnitude of phototaxis increased when changing the light contrast from 0 to 1 (Figure 4.16 D). Contrast was measured as $\frac{I_{max}-I_{min}}{I_{max}+I_{min}}$, where I_{min} and I_{max} are minimum and maximum light intensities respectively. Changing initial light intensity to 50% or to 10% of the original intensity value did not alter the pattern of the response to contrast levels in the tested larvae (Figure 4.16 A). These findings supported the hypothesis derived from the reconstructions of the electron microscopy data and showed that the larvae respond to light contrast rather than light intensity.

4.4. Discussion

Recent advances in imaging techniques resolve anatomical structures on the level of single cells and organelles, which allows the reconstruction of animal neuronal circuitries and subsequent exploration of their function. Computational means largely facilitate such analysis: they provide efficient ways to visualize and navigate through large image datasets, segment and reconstruct anatomical structures and analyze and understand biological processes based on the reconstructed anatomies. Due to the visual nature of electron microscopy data, image analysis techniques are especially useful at each step of the data analysis process.

As part of my project, I developed image analysis tools to study neuronal networks and to understand the relationship between the structure and the function of an animal nervous system. I created software named NeuroDetective for representing and exploring the architecture of biological neuronal networks and the information flow within them. I provided a neuronal model for the public via a web application and an application for Android mobile devices. Furthermore, I developed Fiji plugins to analyze behavioral experiments and verify hypotheses about the neuronal processes modulating animal behavior.

4.4.1. A tool for visualization and analysis of a neuronal network

Reconstructing neuronal networks from electron microscopy images yields neuronal morphology and synaptic connectivity, which are crucial for understanding neuronal network functionality. Software applications for network reconstruction often provide direct access to neuron morphology, for example, via 3D View in TrakEM2 [129], 3D Viewport in Knossos [136] or Structure Viz in Viking [140]. Some tools also provide the 2D connectivity graph of a reconstructed neuronal circuitry, for example, via Circuit Viz in Viking or Graph View in ConnectomeExplorer [132].

Synaptic connectivity and neuronal morphology can be exported from the reconstruction software for subsequent analysis in specialized external tools. The morphology of reconstructed

anatomical structures can be visualized and explored in 3D visualization software (such as Blender [164], Maya [163] or Amira [165]). Another option is the Neuromorph plugin set [180] for Blender, which was designed specifically for visualization and morphometric analysis of the anatomical structures reconstructed from electron microscopy images. Synaptic connectivity can be exported and subsequently explored in various graph analysis tools (such as Gephi [166], BiologicalNetworks [181], MatrixExplorer [182], VOSviewer [183], etc). Various approaches have already been employed for analyzing neuronal connectivity, such as VOSviewer [124], Gephi [119, 12] and Cytoscape [184, 119].

Here, I developed a software application, NeuroDetective, which integrates the morphology and connectivity of network components, thereby combining the advantages of 3D visualization and graph analysis tools. NeuroDetective is not restricted to *Platynereis* and can be adapted to the neuronal network of any organism and tailored to specific research questions. I provided NeuroDetective with convenient instrumentation for storing attributes associated with a neuronal circuitry and accessing these attributes in a straightforward and non-overwhelming manner, a crucial point given the complexity of neuronal networks. Integrating functionality for storing, visualizing and analyzing neuronal network data in a solitary software significantly facilitates visual exploration of neuronal networks.

Neuronal network model

NeuroDetective provides the functionality to model reconstructed neuronal networks in an automatic manner and to represent these networks in a simplified form in Blender. The simplified anatomy omits overwhelming morphological details, allowing the user to focus on the main structures. It also compensates for imaging artifacts and imperfections in the alignment of serial section electron microscopy images. Automatically imported structures might require subsequent curation, for example, in cases where the smoothing function in Blender leads to a separation of the axon branches. Nevertheless, automatic import is an efficient and less biased method compared to hand-drawing.

I incorporated tools to handle synaptic connectivity in the software: to import, store and conveniently access this information. Depending on the approach, incorporating information on numerous synapses in a model can be a labor-intensive and error-prone task. Automatic synapse import significantly reduces the required effort and minimizes errors compared to a manual method.

Synaptic connectivity exploration

NeuroDetective visualizes the spatial distribution of synaptic connections, which reveals anatomical locations with higher information exchange rates and, thus, higher importance for the network functionality. Identifying such regions can guide future experiments involving laser ablations or electric and chemical stimulation and subsequently monitoring of the propagation of neuronal activity.

NeuroDetective explores neuronal connectivity of the elements of a neuronal network and highlights all incoming or outgoing synapses of an individual cell, thereby unveiling the range of the cell's influence. It also visualizes synapses between two cells, which reveals the strength of the connection between those cells and the importance of that cell-cell communication for the network activity. NeuroDetective also highlights the pre- and postsynaptic sites of a cell or focuses on its up- or downstream subnetworks to efficiently explore the information exchange between sensory, inter- and motoneurons and to extract patterns of interactions associated with certain biological processes. Accessing neuronal connectivity information in combination with network anatomy, reveals spatial localization of the information flow within a network relative to anatomical structures.

NeuroDetective extensively studies neuronal circuitry by estimating its statistical properties or by representing network centrality in the context of the neuronal anatomy. Such analysis is crucial for determining influential network elements, strongest connections and information flow critical for biological processes.

4.4.2. Software for visual data analytics

The large scale and extreme complexity of neuronal networks represent a challenge for data analysis and information extraction. Human expert involvement is often necessary to solve this problem. Visual data analytics is a discipline concerned with representing data in a way that enables a human to interact with it, generate hypotheses and draw conclusions [185, 186]. The generated hypotheses can be verified, if necessary, by automatic techniques, experiments or further visual exploration. This approach proves to be effective in extracting essential patterns from large amounts of multidimensional data. Visual data analytics usually follows the path of: "Overview first, zoom and filter, then details on demand", which is known as a visual information seeking mantra [187]. The idea is that an expert first receives a general overview of data and then focuses on a subset of interest, for which additional details can be provided. NeuroDetective provides functionality for visual data analysis following the proposed mantra. **Overview.** NeuroDetective provides a simplified representation of the reconstructed neuronal network. Color-coding by anatomical classification provides the distribution of different classes of the elements in the network. Color-coding by network centrality measure displays informa-

tion flow through all network elements.

Zoom and filter. The software can focus on individual cells or groups of cells categorized by behavior, gene expression and anatomy.

Details on demand. Connectivity, graph centrality and statistical information can be explored for the cells or groups of interest.

Following the guidelines for the visual analytics, NeuroDetective represents a convenient tool for visual network exploration.

4.4.3. Knowledge base of *Platynereis* nervous system

NeuroDetective was applied to study the neuronal circuitry in a 72 hpf *Platynereis* specimen. The reconstructed neuronal networks downstream of the eye photoreceptors and other sensory cells of interest were modeled and subjected to analysis using the software tool. The *Platynereis* neuronal model was extended with anatomical classification, behavior and gene expression, which proved to be useful for the network analysis on the group level.

Accessing network elements according to anatomical classification allowed for a focus on specific cell types and the investigation of the patterns of spatial distribution and relative cell positions, which elucidated novel aspects of the network functionality. NeuroDetective explores synaptic connectivity within and between anatomical groups, discovers connectivity patterns and unravels the information flow on the level of groups (Figure 4.11). For example, the sensory cells (the eye photoreceptors, the ciliary photoreceptors and the mechanosensory cells) showed no or relatively few presynaptic sites and incoming synapses, which suggested relatively low modulation of sensory organs and stimulus sensation. The primary interneurons were highly connected with many synaptic sites, which suggested an important role for the network functionality. In contrast, low connectivity of the ventral interneurons suggested their modest role in the network activity. Equivalent input and output levels for most interneuron types reflect their transmissive role in the network.

Storing and querying gene expression information allowed for the investigation of molecular aspects of biological processes in *Platynereis* larvae, which in conjunction with the neuronal architecture provides an extensive picture of the neuronal circuitry functionality. Several methods were used to obtain gene expression information for the neuronal network model, such as those based on cell morphology, anatomical landmarks, spatial gene colocalization or image registration. Method accuracy largely depends on the specificity of the gene expression and its anatomical localization. One method that enables mapping of gene expression onto a neuronal network more accurately is the combination of electron microscopy sections and gold-immunolabelling. However, this method is limited to a set of genes compatible with the immunochemistry procedure. Immunolabelling of the electron microscopy sections is an on-

going work in *Platynereis*, which will extend the current model with further gene expression information.

The *Platynereis* neuronal circuitry model was annotated with behavioral information obtained from behavioral experiments or based on knowledge of specific cell types and their respective functionality (such as the photoreceptors involved in phototaxis). The *Platynereis* nervous system is far from being fully understood, therefore sparse behavioral annotation will be further extended with subsequent behavioral studies. Annotation of the *Platynereis* neuronal circuitry model incorporates the available data associated with the neuronal network within the single software tool, thereby founding a knowledge base for the *Platynereis* nervous system.

4.4.4. Understanding neuronal circuitry function

The Fiji plugins developed for analyzing behavioral experiments helped to verify hypotheses on network functionality derived from the reconstruction of the *Platynereis* nervous system. A custom plugin was used to analyze behavioral experiments on *Platynereis* phototaxis and to understand the role of the eye photoreceptors in this activity. The plugin automatically analyzed experimental data in an unbiased manner and showed that both adult eyes play crucial role in the light-response behavior of late *Platynereis* larvae.

Another plugin was successfully applied to the experimental data to verify the hypothesis derived from the electron microscopy data about the information flow within the *Platynereis* neuronal circuitry. Contralateral communication between the primary interneurons suggested that the input signals from the opposite sides of the animal are mutually suppressed, thereby enabling larvae to react to light contrast rather than to light intensity. Analyzing the behavioral experiments showed that the *Platynereis* larvae indeed responded to light contrast and therefore supported the hypothesis derived from the neuronal network reconstruction.

The image analysis tools provided here proved to be useful in analyzing behavioral experiments and facilitated our understanding of experimental results, thereby helping to establish the link between the structure and the function of the reconstructed nervous system in *Platynereis*.

4.4.5. *Platynereis* neuronal circuitry for mobile and online exploration

Online NeuroDetective is a web application that was developed for the interactive exploration of the *Platynereis* neuronal circuitry. The advantage of Online NeuroDetective is that it does not require one to install or master any 3D graphics software. The application for Android mobile devices, Mobile NeuroDetective, was also developed to provide a straightforward overview of the *Platynereis* neuronal circuitry model. Importantly, this application does not require an Internet connection and allows the access and interactive exploration of the neuronal circuitry

model offline. Being installed on a mobile device, Mobile NeuroDetective is always available and can be conveniently used during poster presentations or spontaneous scientific discussions.

4.4.6. Synaptic connectivity in other species

The reconstructed neuronal network of *Platynereis* was explored using statistical tools provided in NeuroDetective and was compared to nervous system reconstructions of other model organisms. The model of the neuronal circuit in *Platynereis* has 1143 synapses and 94 neurons categorized into 20 classes. The *Caenorhabditis elegans* (*C.elegans*) nervous system comprises 6393 chemical synapses, 890 gap junctions and 279 neurons divided into 118 classes [117, 120]. A project realized by Takemura *et al.* [124], who reconstructed part of the *Drosophila* brain, contains 8637 synapses and 379 cells falling into 59 classes. Similar to Takemura *et al.*, the *Platynereis* project focuses on certain aspects of the animal's nervous system, thereby concentrating on neuronal mechanisms involved in a particular biological activity. However, the current project is executed on a smaller specimen, taking advantage of the relatively simple organization of the nervous system of the *Platynereis* larva in conjunction with the complex behavior.

The *C.elegans* nervous system is more densely connected than the reconstructed *Platynereis* circuitry: there are 7.86 pre- and postsynaptic sites per neuron on average in *C.elegans* [120] and 3.6 in *Platynereis*. However, *Platynereis* shows slightly stronger connections with 3.82 chemical synapses per connection in *Platynereis* and 2.91 in *C.elegans* [120].

Similar to the reconstructions of the *C.elegans* nervous system and rat visual cortex [126], the *Platynereis* specimen exhibited a large number of weakly connected neurons and few densely connected neurons. This distribution pattern is common for many different types of networks, for instance, in social interactions [188], co-authorship networks [188] or RNA interaction networks [189].

No significant correlation of the number of pre- and postsynaptic sites for the *Platynereis* circuitry was found (0.22), in contrast to the correlation of 0.71 ($p=0.00$) in *C.elegans* [120]. This suggested low interference between input and output strength for the neurons in the *Platynereis* larval nervous system.

4.4.7. Future perspectives

Understanding general patterns in neuronal connectivity

The relatively simple organization of the *Platynereis* nervous system and its low number of neurons allow investigation of the neuronal network patterns and motifs. Neuronal network

modules can re-occur with high probability in other species, due to the ancestral nature of cell types in *Platynereis dumerilii* [43]. Thus, understanding the relatively simple neuronal circuitry in *Platynereis* can facilitate our understanding of more complex neuronal networks in other species. Neuronal connectivity patterns in *Platynereis* can also be compared to those in other organisms and thereby facilitate the study of neuronal network pattern and motif evolution.

The genetic basis of nervous system development can be studied by exploring and comparing the neuronal networks of normal and genetically manipulated individuals. Such studies would be significantly facilitated using NeuroDetective, which provides an efficient platform for investigating neuronal networks.

Bio-inspired technologies

Understanding the entire *Platynereis* nervous system will further extend the generated model, which would enable simulation of the neuronal circuitry and prediction of the response to a given external stimulus. Such a model could be used to replicate larval behavior in natural habitats and investigate behavioral responses to change in various environmental parameters, such as temperature or pH value. Since *Platynereis* larvae are a part of marine plankton, such simulations can be valuable for ecological studies.

Comprehending the neuronal processes in *Platynereis* can also be beneficial for developing bio-inspired systems, such as that developed by Schmuker *et al.* [190], who were inspired by the insect olfactory system and implemented a software-hardware system solving a classification task based on this system.

The existing bio-inspired systems can also be improved, for instance, the Breitenberg vehicle, a hardware system able to react to a light stimulus [191]. The vehicle is equipped with two light sensors directly connected to motors, thereby representing a simple sensorimotor system able to interact with an environment and react to a stimulus. Depending on the connection pattern, the vehicle demonstrates different behaviors (for example, approaching or avoiding the light source). Employing mechanisms of stimulus response in *Platynereis*, the Breitenberg vehicle can be extended with additional units, such as "primary interneurons", to be able to respond to a stimulus in a more complex manner.

4.4.8. Conclusion

NeuroDetective was developed to enable the interactive exploration of complex neuronal circuitries and analysis of their statistical and network properties. NeuroDetective was applied

to study the flow of electrochemical activity in the neuronal circuitry of a 72 hpf *Platynereis* specimen and to unveil neuronal mechanisms modulating its behavior. This neuronal circuit was annotated with anatomical classification and behavioral and gene expression information, thereby providing a knowledge base for *Platynereis* comprising comprehensive information associated with the neuronal network. NeuroDetective proved to be suitable for analyzing biological information flow within a neuronal network and for discovering mechanisms of interactions between neuronal units crucial for realizing the biological activities of an organism. Analyzing the *Platynereis* neuronal network in NeuroDetective shed new light on the evolution and development of the nervous system and its behavioral capacities. Neuronal network analysis revealed crucial principles of biological information processing, which can potentially guide further experimental manipulation or inspire the development of more efficient artificial intelligence architectures.

5. Integrating light and electron microscopy datasets

5.1. Introduction

Connectomics is one of the active fields in neuroscience that aims at assembly and functional study of large neuronal circuits within the nervous system of an organism [124, 127, 192]. Electron microscopy (EM) is the technique widely used in connectomics to resolve neuronal structures and their interconnections. The challenge in understanding brain function is the flexibility of neuronal circuits, in other words, the same neuronal circuit can be responsible for multiple functions [192]. Switching between neuronal circuitry activities is modulated by hormones, neurotransmitters or peptides. Localizing these modulators relative to the neuronal circuitry architecture can elucidate the mechanisms of switching between different activities and reveal the functional repertoire of a neuronal circuitry. Staining techniques (e.g. immunochemistry or *in situ* hybridization) combined with imaging methods are widely used to localize biomolecules within an organism. Peptidergic modulators can be localized using specific antibody markers, whereas modulating hormones, or their activity sites, can be localized by determining their receptors and using molecular markers for those receptors.

Light microscopy (LM) is used to localize the signal of biomolecule markers, however, the resolution of this modality is limited to the light wavelength, which hinders the resolution of fine details. In contrast, the EM modality resolves subcellular structures; therefore it was natural to combine both modalities to solve the problem: correlative light and electron microscopy (CLEM) incorporated the advantages of the LM and EM modalities by matching molecular marker signal from LM to the morphological details obtained from EM. CLEM has often been employed to investigate intracellular processes, such as Golgi-to-membrane transport [193], RNA translocation [194], chromosome dynamics [195, 196] or virus replication in a host cell [197]. The major challenge in CLEM is to align images acquired from two modalities. Adding landmarks detectable by both LM and EM to a specimen facilitates signal matching. A find grid [198], sometimes coated with gold or carbon [199], engraving embedment substrate [200] or laser carving [201] can provide landmarks for alignment.

Due to the difference in specimen preparation for LM versus EM imaging and consequent

heterogeneous tissue deformations, whole organisms represent a more challenging task for CLEM. Despite this, CLEM has been successfully applied to understand whole-body processes in such organisms as zebrafish [202], *Caenorhabditis elegans* [203] and *Drosophila melanogaster* [204]. CLEM can start with obtaining *in vivo* signal using LM modality (for instance, employing genetically encoded markers), followed by the preparation of a specimen and imaging using the EM modality [202]. It has also been shown that auto-fluorescent signal can be preserved after processing for EM in some species [205], thus, CLEM can start with preparing a specimen and then imaging it using both LM and EM modalities.

Another technique to obtain active sites for biomolecules in the context of anatomical ultrastructures is immunocytochemistry on EM sections [206]. Gold particles coupled to antibodies are widely used as efficient markers, as they are electron dense and consequently detectable in EM images [207, 208, 209]. Another advantage of gold particles is that they can be produced in different sizes and can therefore be used with various antibody markers in order to detect several different biomolecules on the same section. The success of immunocytochemistry on EM sections depends on the specificity, avidity and robustness of the markers. Gold immunostaining has been successfully applied to rat hippocampus [210], mouse and rat brains [201].

An alternative approach to bring together LM and EM modalities is to use image registration. Image registration on the differently prepared and, thus, differently deformed specimens is challenging. No universal automatic solution to this problem has been discovered so far. Cardona *et al.* [211] proposed an approach based on user-defined landmarks and extended TrakEM2 with 3D non-linear registration of EM and LM image stacks based on specified landmarks.

Here, I propose a method to correlate signals obtained from LM and EM imaging modalities based on automatic image registration. The method was applied to *Platynereis dumerilii*, which exhibits stereotypic development in its early developmental stages and is therefore amenable to image registration. However, distinct preparation procedures for EM versus LM imaging resulted in drastic morphological differences in the prepared specimens. The presented method largely overcomes the differences and therefore enables the exploration of neuronal circuits in conjunction with molecular modulators and the investigation of the function repertoire of the neuronal circuitry of interest.

5.2. Methods

5.2.1. Image set from electron microscopy

Electron microscopy (EM) was performed as described in [12]. The obtained images were imported into TrakEM2 [129], where image tiles from a single layer were stitched and the stitched

layers were aligned into a stack. The images were then exported from TrakEM2 using the *Export flat images* option with the scale of 0.05 (Figure 5.2). All subsequent image-processing operations were performed in Fiji [101]. The median filter in z-direction was applied to the exported images in order to remove imaging artifacts in individual layers. The white edges resulting from the epon embedding were removed with an automatic algorithm implemented in Fiji using the ParticleAnalyzer plugin. For visualization purposes, the image intensity was inverted to match the intensity of the light microscopy (LM) template. The images were scaled down in three directions by a factor of 3 using Gaussian filter with sigma 2.5 and again by a factor of 3 in the Z-direction. After rescaling, Gaussian filter was again applied to ensure smooth intensities in the stack.

5.2.2. Registration of the light and electron microscopy images

Landmark detection The processed EM image stack was further adjusted to prepare for registration. Every third layer within the first 100 layers (starting from the apical part) was removed to better match LM and EM datasets. The following 70 layers (from 101 to 170) of the stack remained intact (Figure 5.1 A). The lipid drops (Figure 5.1 B) present in the layers starting from 171 to the end were roughly segmented out (Figure 5.1 C).

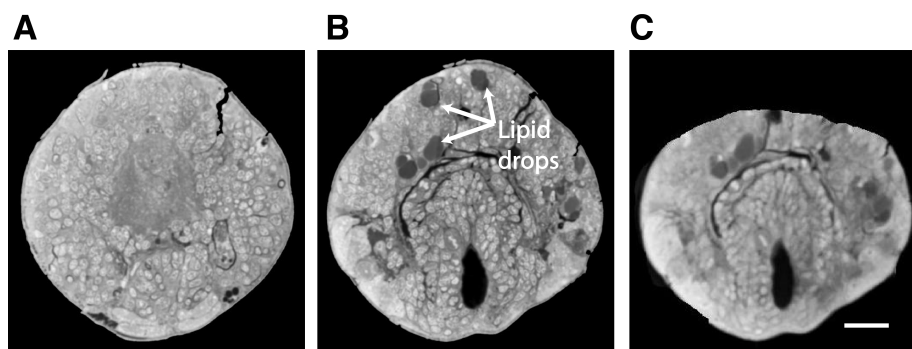


Figure 5.1.: Adjusting electron microscopy stack for registration.

(A) A sample layer from the electron microscopy stack on the level of plexus. (B) A sample layer from the electron microscopy stack containing lipid drops. (C) A sample layer from (B) roughly segmented. Scale bar 20 μm .

Image set from light microscopy The DAPI-based average template, which was used for generating the gene expression atlas (cf. Gene expression atlas for *Platynereis dumerilii*), was oriented apically. The layers starting from the level roughly corresponding to the last image of the EM stack (below the ciliary band) were removed. The gene expression patterns, registered to the DAPI-based template as described in Gene expression atlas for *Platynereis dumerilii*, were oriented apically and cut at the same level as the template.

Image registration The registration procedure was similar to that described in the Chapter "Gene expression atlas for *Platynereis dumerilii*". Affine and deformable registration was applied to register the modified LM template to the modified EM image stack using the ITK toolkit. The registration was performed once and the obtained transformation was reapplied to the gene expression patterns from the LM dataset.

5.3. Results

Applying different imaging modalities to a specimen of interest reveals different aspects of biological processes occurring in that specimen. *In situ* hybridization combined with the light microscopy (LM) imaging localizes gene expression patterns, thereby revealing functional information for a labeled region. In contrast, electron microscopy (EM) imaging unravels detailed morphology of individual neurons and localizes the synaptic connections between neurons, thereby providing the structure and connectivity of neuronal circuits. Combined, gene localization and nervous system architecture provide extensive information for understanding the processes in a nervous system modulating specific biological activities. It is therefore important to localize gene expression patterns in the context of detailed anatomical structures. Here, I present a procedure to localize gene expression patterns obtained using LM imaging in the context of the neuronal anatomy reconstructed from the EM images. The method was applied to *Platynereis* larvae at 72 hours post-fertilization (hpf).

5.3.1. Modification of the electron microscopy stack

First, I prepared the LM and EM image sets of *Platynereis dumerilii* for registration (cf. Methods). The EM image stack resembles a 72 hpf specimen scanned using EM. The images were imported into TrakEM2, where they were stitched and aligned, and subsequently exported as an image stack (Figure 5.2). The exported stack was rescaled, processed using median and Gaussian filters and segmented in order to obtain the size and appearance similar to the LM template. The LM dataset consisted of the average DAPI-template previously used for generating the gene expression atlas (cf. Gene expression atlas for *Platynereis dumerilii*). The template was oriented apically and the layers below the ciliary band were removed. The average gene expression patterns registered to the LM template (cf. Gene expression atlas for *Platynereis dumerilii*) were subjected to the identical procedure.

An attempt to register the LM template directly to the EM stack did not result in an appropriate alignment: though the boundaries of the registered larval bodies were aligned, internal anatomical structures, such as mouth or apical neural plexus, were largely mismatched. This was due to the prominent differences in the tissue deformations caused by the preparation pro-

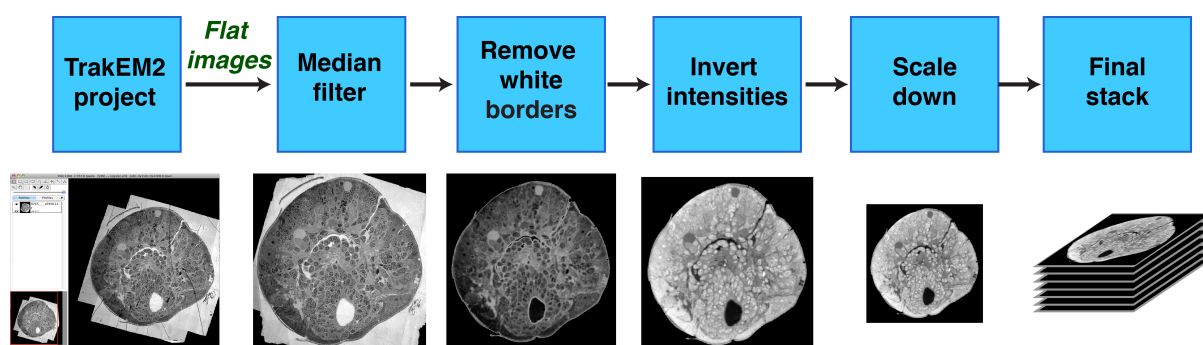


Figure 5.2.: The pipeline for preparing electron microscopy images for registration. The images were first exported from the TrakEM2 project as a set of 'flat images'. The median filter in z direction was applied. The white borders from the epon embedding were automatically removed. Image intensities were inverted to match the intensities of the light microscopy template. The image stack was scaled down in three directions.

cedures. Visual inspection of the registration results revealed that the lipid droplets in the EM stack (absent in the LM template) and the disproportional deformation of the tissues were likely responsible for the misalignment (Figure 5.3).

To solve this problem, I further modified the EM stack to maximize similarity between the datasets. Each third layer in the apical part of the EM stack (layers 1-100) was removed to obtain proportions similar to that of the LM stack (Figure 5.1 2 A). To eliminate the lipid droplets, I roughly segmented the layers manually using TrakEM2 (Figure 5.1 B, C). Segmentation was iteratively adjusted so that the tissue surrounding mouth did not distort drastically during the registration.

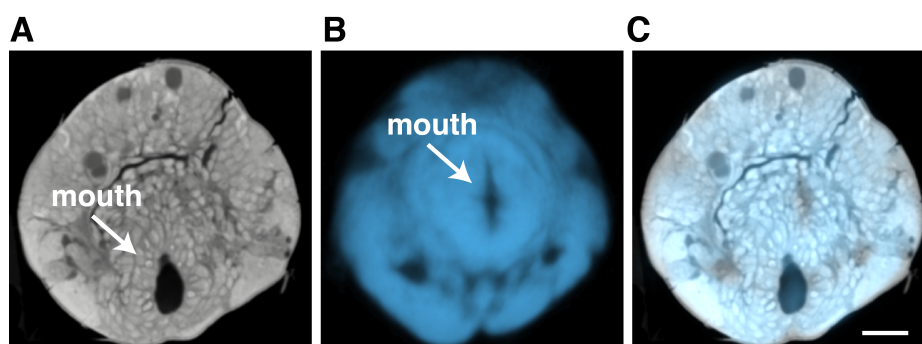


Figure 5.3.: Preliminary registration of the light microscopy template to the electron microscopy stack. (A) A layer from the electron microscopy stack. (B) A layer from the light microscopy template registered to the electron microscopy stack. (C) A layer from the overlaid electron microscopy stack and the registered light microscopy template. Scale bar $20\ \mu\text{m}$.

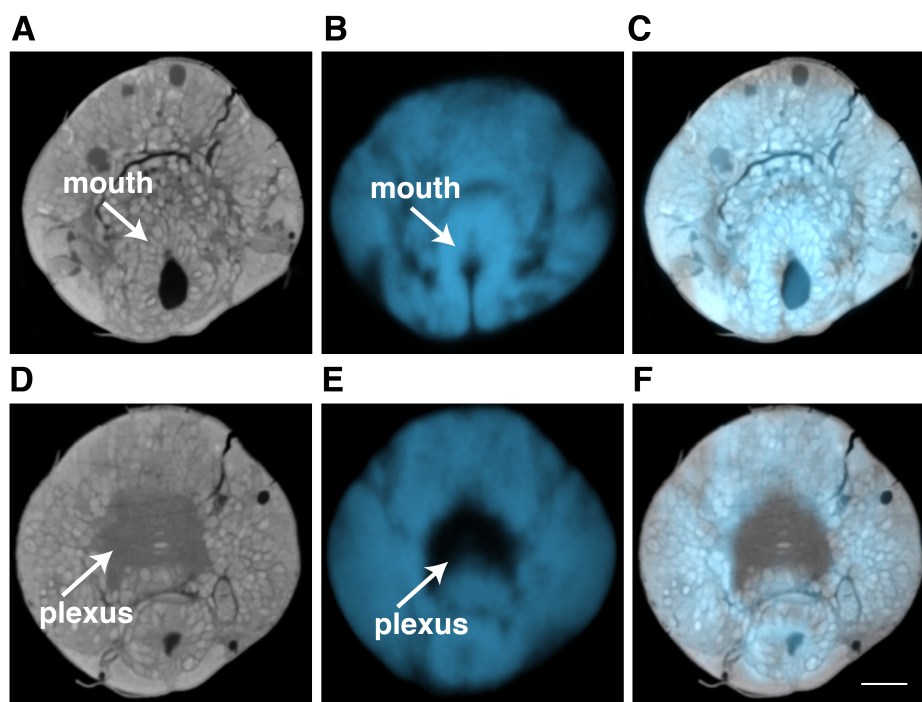


Figure 5.4.: Registration of the light microscopy template to the adjusted electron microscopy stack.

(A, D) Individual layers from the adjusted electron microscopy stack at the level of the mouth (A) and the apical neural plexus (D). (B, E) Individual layers from the registered light microscopy template at the level of the mouth (B) and the apical neural plexus (E). (C, F) Individual layers from the overlaid electron and light microscopy stacks at the level of the mouth (C) and the apical neural plexus (F). Scale bar 20 μm .

5.3.2. Light microscopy signals mapped to the electron microscopy stack

To align the LM template to the EM stack, I employed the image registration procedure described in "Gene expression atlas for *Platynereis dumerilii*". Visual inspection demonstrated satisfying registration results: internal structures were closely matched, the body boundaries revealed appropriate alignment and no unnatural distortions were detected (Figure 5.4).

The registration yielded the transformation that projects information from the space of the LM template to the space of the EM stack. The transform was applied to the gene expression patterns from the LM dataset. First, I applied the transformation to the tubulin staining, which labels neuronal processes and cilia in the *Platynereis* larva (Figure 5.5 A). Visual inspection of the results revealed close matching of internal structures in the LM and EM datasets (Figure 5.5).

Next, I applied the transformation to the *Tinman* gene expression pattern (Figure 5.6). From the *in situ* hybridization experiment, it is known that *Tinman* is expressed in the region around the mouth in *Platynereis* larvae (Figure 5.6 D). *Tinman* expression pattern registered to the EM space was also located around the mouth in the EM stack (Figure 5.6 E).

Registration should bring all corresponding anatomical structures together, therefore the accu-

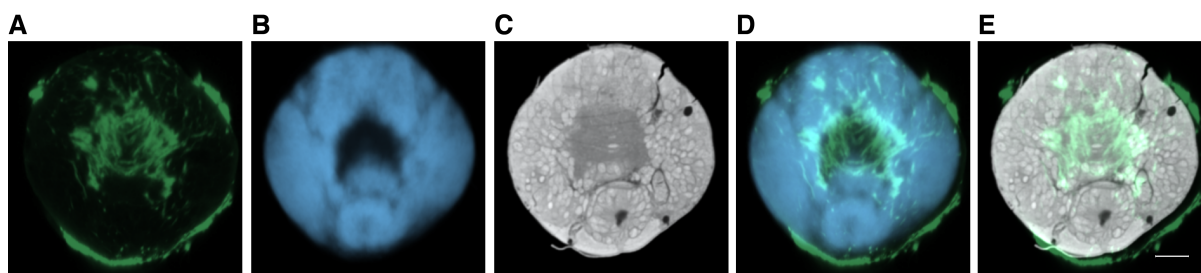


Figure 5.5.: The tubulin signal registered to the electron microscopy stack.

(A) The registered tubulin staining, an individual layer is shown. (B) The registered light microscopy template, an individual layer is shown. (C) A layer from the electron microscopy stack. (D) The overlaid tubulin staining and the light microscopy template, an individual layer is shown. (E) The overlaid tubulin staining and the electron microscopy stack, an individual layer is shown. Scale bar $20\ \mu\text{m}$.

racy of the method can also be quantified based on the positions of the corresponding anatomical structures after registration. Ciliary photoreceptors were selected as anatomical structures for evaluation, since they can be detected in the LM dataset based on tubulin staining (Figure 5.7 A, D) and the EM dataset based on their distinctive morphology (Figure 5.8 A, B). The ciliary photoreceptors were segmented from the registered tubulin staining in the LM and EM datasets and their positions were compared. The distance was $2.05\ \mu\text{m}$ between the centers of the corresponding ciliary photoreceptors on the left side and $0.46\ \mu\text{m}$ for the ciliary photoreceptors on the right side (Figure 5.7).

I also quantified the quality of the registration using the *Platynereis* adult eyes as landmarks. The adult eyes resemble a group of cells where *rhabdomeric opsin-1* (*r-opsin-1*) is expressed, therefore they can be localized based on the *r-opsin-1* expression pattern in the LM dataset (Figure 5.9 B, E) and based on morphological landmarks in the EM dataset (Figure 5.8 C, D). The registration transform was applied to the *r-opsin-1* expression pattern from the LM dataset. The centers of mass for the *r-opsin-1*-expressing cells were compared to the centers of mass of

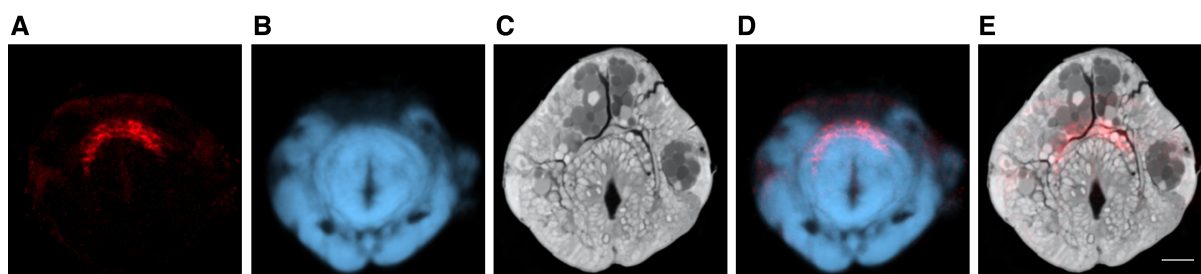


Figure 5.6.: Tinman gene expression pattern registered to the electron microscopy stack.

(A) A layer from the registered stack containing *Tinman* expression pattern. (B) A layer from the registered light microscopy template. (C) A layer from the electron microscopy stack. (D) A layer from the overlaid registered *Tinman* stack and the registered light microscopy template. (E) A layer from the overlaid registered *Tinman* stack and the electron microscopy stack. Scale bar $20\ \mu\text{m}$.

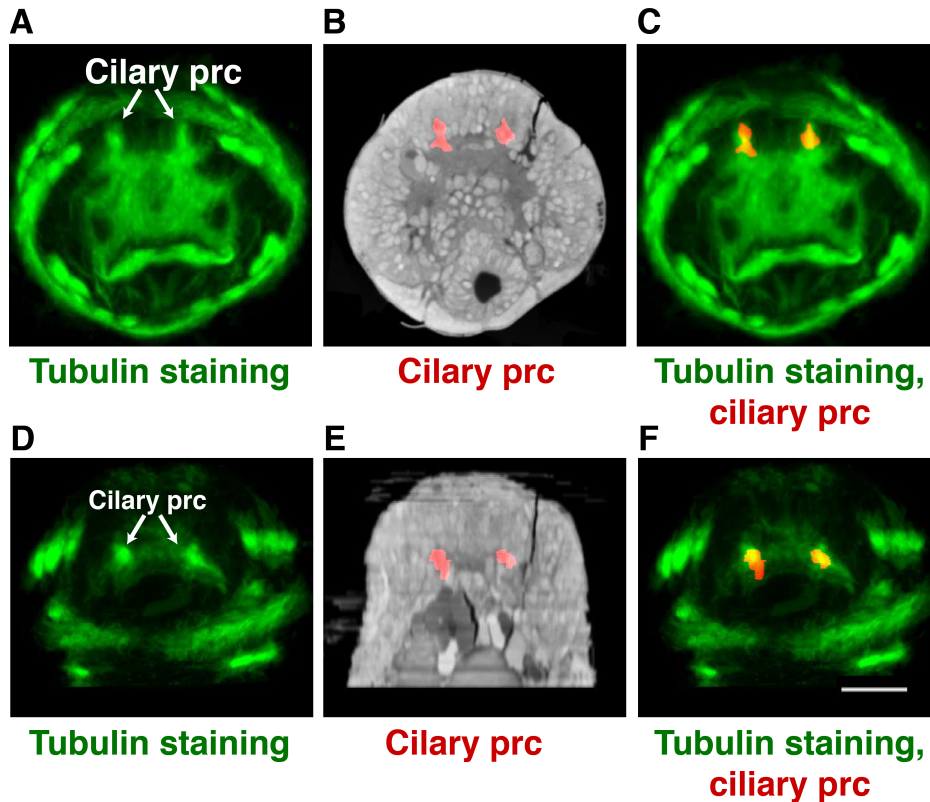


Figure 5.7.: Distance between ciliary photoreceptors defined in the electron and registered light microscopy stacks.

(A, D) The ciliary photoreceptors (indicated with arrows) defined in the tubulin staining registered to the electron microscopy stack. Apical (A) and ventral (D) views are shown. (B, E) The ciliary photoreceptors defined in the electron microscopy stack based on cell anatomy. Apical (B) and ventral (E) views are shown. (C, F) The overlaid tubulin signal and segmented ciliary photoreceptors. Apical (C) and ventral (F) views are shown. Scale bar $30\ \mu\text{m}$. prc, photoreceptor cells.

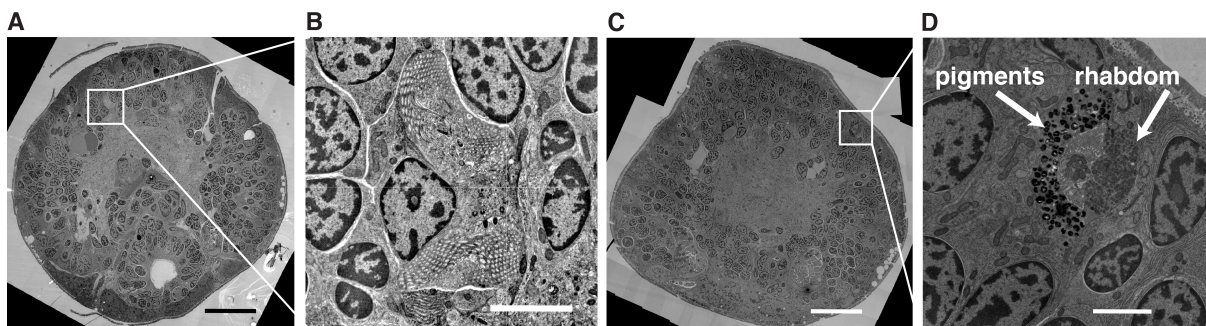


Figure 5.8.: The ciliary and visual photoreceptors defined in the electron microscopy images. (A) An individual electron microscopy section with a ciliary photoreceptor shown. (B) A close-up of a ciliary photoreceptor in the electron microscopy image. (C) An individual electron microscopy section with a visual photoreceptor shown. (D) A close-up of a visual photoreceptor in the electron microscopy image. Scale bar $20\ \mu\text{m}$ in A and C, $2\ \mu\text{m}$ in B and D.

the visual photoreceptors segmented from the EM stack. The distance was $11\ \mu\text{m}$ between the rhabdomeric photoreceptors on the left side and $33\ \mu\text{m}$ for the rhabdomeric photoreceptors on the right side (Figure 5.9).

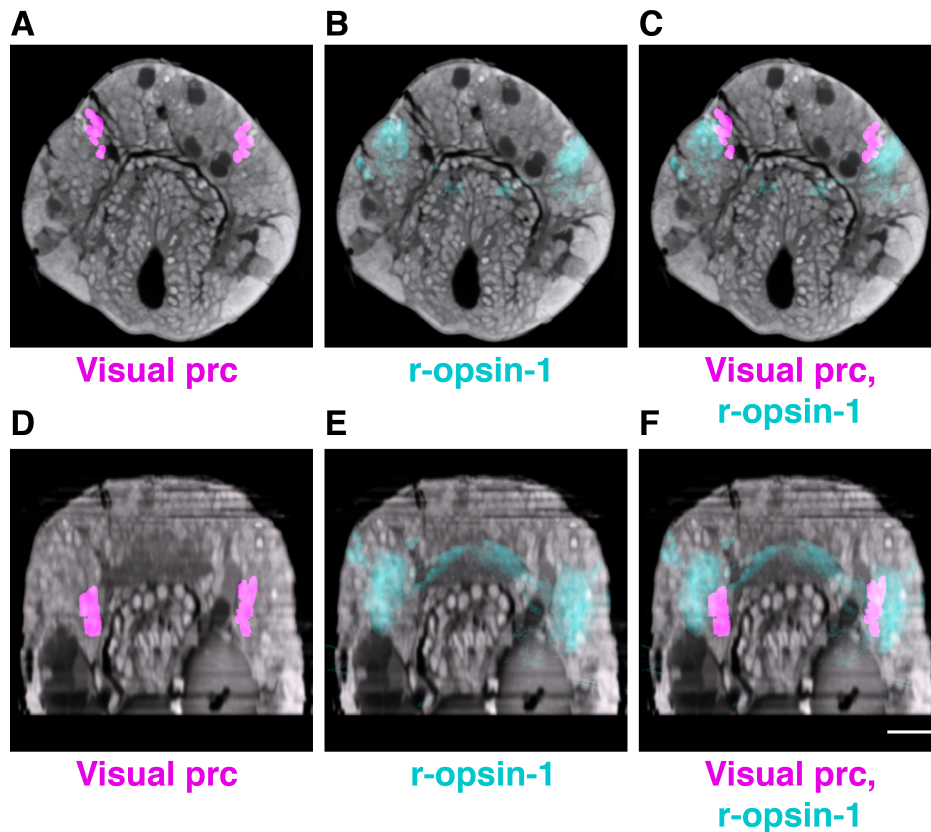


Figure 5.9.: Distance between the visual photoreceptors in the electron and light microscopy stacks.

(A, D) The rhabdomeric photoreceptors defined in the electron microscopy stack based on cell anatomy. Apical (A) and ventral (D) views are shown. (B, E) The *r-opsin-1* expression pattern defining rhabdomeric photoreceptors. Apical (B) and ventral (E) views are shown. The overlaid the *r-opsin-1* expression pattern and the rhabdomeric photoreceptors segmented in the electron microscopy stack. Apical (C) and ventral (F) views are shown. Scale bar $20\ \mu\text{m}$.

5.4. Discussion

I presented an intensity-based registration method for correlating signals from two imaging modalities: light and electron microscopy. The method was applied to *Platynereis dumerilii* larvae at 72 hours post-fertilization (hpf). The registration results demonstrated a close match of the internal structures detectable in both image sets.

Due to the differences in specimen preparation for the light versus electron microscopy imaging, registration represents a challenging task. Since the most prominent signal for registration

was provided by the apical nuclei, the apical neural plexus and the mouth, those regions yielded the most accurate results. In contrast, structures located further from the main landmarks display poorer results, which explains the rather large distance between the registered and segmented photoreceptors. However even poor results can provide valuable assistance in defining corresponding regions in the LM and EM datasets, as it suggests a group of candidate cells, which can be subsequently resolved based on neuronal morphology, anatomical landmarks or specific cell features.

Previously, the problem of correlating signal from LM and EM has been approached using various technical and computational techniques. A technical solution has been provided by correlative light electron microscopy (CLEM), which images fluorescently tagged proteins using LM and visualizes internal structures via EM. However, CLEM is restricted when applied to whole organisms, since the morphology of a specimen prepared for imaging using LM can differ considerably from a specimen prepared for EM. The image registration method presented here aims to align the images of whole organisms by compensating for differences in tissue deformation using affine and deformable registration.

Another technical solution to match LM and EM signals was achieved via immunolabeling of the EM sections. As light microscopy imaging can be performed on the thin sections prepared for EM imaging, tissue deformation is not a limitation for this technique. Nevertheless, immunolabeling of the EM sections is restricted in terms of the robustness and the specificity of the antibody markers applied to EM sections. The antibody markers designated for an LM-prepared specimen might not be as effective on EM sections. In contrast, the image registration method presented here is applicable to all markers available from the LM dataset.

Cardona *et al.* [211] proposed a computational approach for matching EM and LM datasets and provided a plugin for registering image stacks based on user-specified landmarks. However, registration approaches based on manually defined landmarks often miss the rich information provided by image intensities. At the same time, highly distorted and dissimilar tissues would require numerous landmarks in this approach. In contrast, the intensity-based registration method employs information provided by image intensities and reduces the requirement for manual input for registering LM and EM datasets.

The presented method integrates nervous system architecture and biomolecular marker signals and consequently allows the investigation of gene functionality in the context of neuronal anatomy or understanding neuronal networks equipped with knowledge of their molecular modulators. The method facilitates exploring regulatory mechanisms responsible for alternating activities of an individual neuronal circuitry and provides a comprehensive picture of biological processes that modulate animal behavior.

6. Discussion

In the course of the current project I developed a platform for analyzing animal behavior and understanding the genetic and nervous system mechanisms modulating this behavior. I employed various image analysis, machine learning and computer graphics techniques for the purposes of the project.

First, I employed image preprocessing, segmentation and tracking to quantify *Platynereis* locomotion and subsequently study mechanisms of phototaxis and the role of adult eyes in this activity. Later, I explored the effect of individual neuropeptides on *Platynereis* locomotion using image preprocessing (segmentation and tracking) and machine learning (Bayesian classification) techniques. The later project aimed at classifying and characterizing swimming and crawling behaviors in *Platynereis*.

Image registration was a key technique for generating the gene expression atlas for *Platynereis dumerilii*. Gene expression atlases overcome the restrictions of labeling and imaging techniques limited in the number of channels and thus, genes. The *Platynereis* gene expression atlas enabled the exploration of the expression patterns of genes involved in larval locomotion, settlement and visual stimulus transduction.

Computer graphics and graph theory provided the basis for the development of NeuroDetective, a tool for exploring the mechanisms of animal nervous systems. NeuroDetective models the reconstructed neuronal circuitry of *Platynereis* and provides extensive tools for the analysis of various aspects of neuronal architecture and connectivity. Use of NeuroDetective gave insight into the biological information flow within the reconstructed neuronal circuitry involved in the response behavior to visual and mechanical stimulus. Additionally, NeuroDetective facilitated the identification of cells and connections important for neuronal circuitry functionality.

Image processing and image registration allowed the combination of gene expression information with the reconstructed neuronal architecture. This advance enabled the study of *Platynereis* neuronal networks in conjunction with their molecular modifiers or the exploration of gene functionality in the context of fine anatomical structures.

The libraries for visualizing 3D models on Android devices and in Web-applications enabled sharing of the *Platynereis* gene expression atlas and the reconstructed neuronal circuitry with the community, which allowed accessible and straightforward exploration of the obtained data.

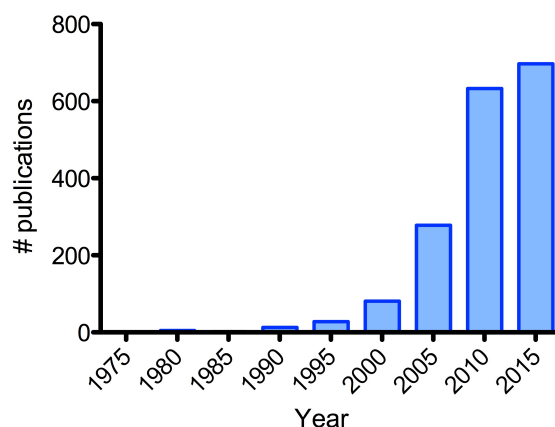


Figure 6.1.: Pubmed statistics of publications concerning bioimage analysis.

The increase in the number of publications over time grouped in periods of 5 year. The search expression used was "biology[MeSH Terms] AND ((computer assisted image analysis[MeSH Terms]) OR (computer assisted image processing[MeSH Terms]))".

Bioimage informatics

The available imaging techniques unravel mechanisms modulating biological micro-processes on the level of organisms, cells or subcellular structures. Advances in imaging techniques in recent years in terms of resolution, variety of signals captured, scanning time and other aspects, allow the obtainment of large volumes of images which can be used to extensively explore biological processes. Computational image analysis techniques significantly facilitate information extraction, which can be challenging due to the amount and complexity of images. Bioimage analysis became a topic for the annual meeting (Bioimage informatics Annual meeting [212]), a devoted Nature Methods issue [213] and numerous publications in such journals as Bioinformatics [214] and BMC Bioinformatics [215], where specialized sections devoted to bioimage analysis were introduced. The number of research projects involving bioimage analysis grows steadily (Figure 6.1). Analysis of biological images grew into a separate field called Bioimage informatics [216, 217], which deals with particular tasks [218], such as:

- guiding image acquisition,
- optimizing image representation and storage,
- overcoming limitations of labeling and imaging techniques,
- extracting meaningful information from microscopic images,
- comprehensively visualizing results

The tools developed in the course of this project deal with the above-mentioned tasks. The generated gene expression atlas for *Platynereis* overcomes the limitations of labeling and imaging techniques by combining multiple gene expression patterns from individual specimens. The

gene atlas facilitates understanding of the obtained gene expression data, by depicting the relative spatial distribution of multiple gene expression patterns and identifying candidates for coexpression and possible interaction. Computationally efficient representation and comprehensive visualization of the gene atlas have been achieved using a model generated in 3D graphics software Blender.

NeuroDetective deals with the task of data representation and storage optimization, representing *Platynereis* neuronal circuitry in the form of a 3D model. NeuroDetective represents large volumes of reconstruction data in a simplified manner thereby reducing storage space requirements. By integrating neuronal morphology and synaptic connectivity information, NeuroDetective provides comprehensive data visualization. The provided functionality for analyzing neuronal circuitry facilitates information extraction. Through the integration of information from the gene expression atlas and neuronal reconstructions, I aimed to overcome the limitations of current imaging techniques available for *Platynereis* by combining information from two different imaging modalities.

Bioimage Informatics is currently facing a number of field-specific challenges. First, the majority of the Bioimage Informatics software users are biologists with little or no programming skills. This imposes particular requirements on the software and its user-interface [219]. Second, some software tools that have been developed are not available to other research groups requiring similar analysis. Unavailability of software leads to another problem - research groups independently develop image analysis tools equivalent to existing tools - leading to inefficient resource usage. Therefore, increasing communication between research groups and collaboration in software development is highly required. However, even when communication and collaboration between research groups are established, problems still can arise concerning the sharing of data. For example, different image formats, diverging image acquisition methods or large data volumes. Standard sharing methods, acquisition protocols, image formats and data representation need to be established in order to move the field of Bioimage Informatics forward in a productive manner.

When developing software tools in the course of the current project, I tried to take into account the problems in the field. All the developed tools are available for the public together with detailed instructions for use. The generated models for the gene expression atlas and neuronal circuitry enabled efficient data sharing via Blender source file or via Web and Android applications. Standard image formats were used throughout the project and existing libraries and techniques were employed and enhanced where possible, thereby ensuring efficient resource usage. Collaborations with other research groups working on *Platynereis* were established to join efforts for creating a public repository for gene expressions.

Contributions

All the research approaches and software tools presented were discussed with my supervisor Dr. Gáspár Jékely leading to new ideas and research directions.

Chapter 1 Gáspár Jékely designed the study. I performed confocal microscopy, implemented the image registration protocol, evaluated the method, generated the gene expression atlas and provided the visualization and analysis tools. I developed the algorithms for behavior analysis. Preparation of images for the gene expression atlas, including *in situ* hybridization and confocal microscopy, was performed by Markus Conzelmann, Elizabeth Williams, Aurora Panzera, Csaba Verasztó, Nadine Randel, Luis Bezares and other former and current members of the Jékely group. Immunocytochemistry was performed by Markus Conzelmann. Nadine Randel performed the behavioral experiments.

Chapter 2 Gáspár Jékely and myself designed the study. I implemented the NeuroDetective platform, evaluated it and developed the tools for behavior analysis. I created the 3D model of the neuronal network and performed the analysis. Reza Shahidi and Nadine Randel performed electron microscopy. Nadine Randel, Luis Bezares, myself and other former and current members of the Jékely group processed the images. Image segmentation was performed by Nadine Randel, Csaba Verasztó, Luis Bezares, Gáspár Jékely and other former and current members of the Jékely group. Nadine Randel and Gáspár Jékely performed behavioral experiments.

Chapter 3 Gáspár Jékely and myself designed the study. I implemented the image registration method, segmented anatomical structures from the electron microscopy images, evaluated the method and applied it to the available gene expression patterns. The light microscopy data from the Chapter 1 was utilized. Nadine Randel, Luis Bezares, Reza Shahidi and myself processed the electron microscopy images.

Publications

- [1] Markus Conzelmann, Sarah-Lena Offenburger, Albina Asadulina, Timea Keller, Thomas A. Muench, and Gaspar Jekely. Neuropeptides regulate swimming depth of *Platynereis* larvae. *Proceedings of the National Academy of Sciences*, 108(46):E1174–E1183, 2011.
- [2] Albina Asadulina, Aurora Panzera, Csaba Veraszto, Christian Liebig, and Gaspar Jekely. Whole-body gene expression pattern registration in *Platynereis* larvae. *EvoDevo*, 3(1):27, 2012.
- [3] Markus Conzelmann, Elizabeth A. Williams, Sorin Tunaru, Nadine Randel, Reza Shahidi, Albina Asadulina, Juergen Berger, Stefan Offermanns, and Gaspar Jekely. Conserved mip receptor-ligand pair regulates *Platynereis* larval settlement. *Proceedings of the National Academy of Sciences*, 110(20):8224–8229, 2013.
- [4] N. Randel, A. Asadulina, L. A. Bezares-Calderon, C Veraszto, R. Shahidi, and G. Jekely. Neuronal connectome of a sensory-motor circuit for visual navigation. *Elife*, 2014.

List of Figures

2.1. <i>Platynereis</i> life cycle	13
3.1. Scanning electron microscopy and light microscopy images of the <i>Platynereis</i> larvae.	22
3.2. A pipeline for template generation.	24
3.3. Gene expression pattern registration pipeline	25
3.4. Exporting gene expression patterns in Blender.	26
3.5. The interface implemented for the model.	27
3.6. Class diagram for the Mobile BrainDiscoverer application.	28
3.7. <i>Platynereis</i> locomotion classification and characterization.	29
3.8. Evaluation of non-rigid image registration.	32
3.9. Visual inspection of the registration results.	33
3.10. Generated templates based on DAPI staining and acetylated tubulin.	34
3.11. Evaluation of the accuracy of the gene expression pattern registration to the DAPI and tubulin templates.	35
3.12. Effect of sample size on average gene expression pattern.	36
3.13. Registration of immunostained specimens to the <i>in situ</i> hybridization template.	37
3.14. Locomotion analysis of the FMRFa-treated larvae.	38
3.15. Gene expression atlases modeled in 3D graphics software Blender.	40
3.16. Gene coexpression analysis in Blender.	41
3.17. Mobile GeneExpressionAtlas.	41
3.18. Online GeneExpressionAtlas.	42
3.19. Colocalization of TRPc and r-opsin-1 expression patterns.	43
4.1. A pipeline for creating a 3D model of a neuronal network.	56
4.2. NeuroDetective user interface.	57
4.3. Schematic importing of synaptic connectivity in a model.	58
4.4. Analysis of the phototaxis experiments.	63
4.5. NeuroDetective functionality for exploring synaptic connectivity.	66
4.6. Adult eyes and mechanosensory cells in the electron microscopy images.	67

4.7. Elements of <i>Platynereis</i> neuronal circuitry reconstructed from the electron microscopy images.	68
4.8. Connectivity matrices for the anatomical groups.	69
4.9. Information flow within the visual circuitry in <i>Platynereis</i> larva.	70
4.10. Distribution of neurons by the number of connections and synapses.	71
4.11. Distribution of connections and synapses by anatomical groups.	71
4.12. Centrality measure for <i>Platynereis</i> neuronal network	72
4.13. Mobile NeuroDetective.	75
4.14. Online NeuroDetective, a web application for exploring the neuronal network.	76
4.15. Eye-ablation experiment analysis.	77
4.16. Light-contrast phototaxis experiment.	78
5.1. Adjusting electron microscopy stack for registration.	89
5.2. The pipeline for preparing electron microscopy images for registration.	91
5.3. Preliminary registration of the light microscopy template to the electron microscopy stack.	91
5.4. Registration of the light microscopy template to the adjusted electron microscopy stack.	92
5.5. The tubulin signal registered to the electron microscopy stack.	93
5.6. Tinman gene expression pattern registered to the electron microscopy stack.	93
5.7. Distance between ciliary photoreceptors defined in the electron and registered light microscopy stacks.	94
5.8. The ciliary and visual photoreceptors defined in the electron microscopy images.	94
5.9. Distance between the visual photoreceptors in the electron and light microscopy stacks.	95
6.1. Pubmed statistics of publications concerning bioimage analysis.	98

List of Tables

3.1. Evaluating the intensity- and landmark-based methods for rigid registration. . .	31
4.1. Software for processing and analyzing electron microscopy data.	52
4.2. Anatomical classification of the elements in the <i>Platynereis</i> neuronal circuitry. .	74

A. Supplementary files

Supplementary files provided in digital form

1. Files, programs, models and instructions to generate and analyze the gene expression atlases are provided:

- a) ImageJ/Fiji programs to rigidly orient images and the instructions on their usage them. These files were also provided as additional files for the online version of the publication [74]: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3586958/>
- b) Programs, scripts and instructions for non-rigid registration using ITK toolkit. These files were also provided as additional files for the online version of the publication [74]: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3586958/>
- c) The average anatomy templates for two-, three- and six-day-old *Platynereis* larvae. The templates are shown in Figure 3.10, and their evaluation in Figure 3.11 G-I.
- d) The raw gene expression patterns and the gene expression patterns registered to the average anatomy templates. The registered gene expression patterns were shown in Figure 3.11 A-F.
- e) The Blender models of the gene expression atlases of *Platynereis* at early developmental stages. The models can be also downloaded from GitHub: <https://github.com/albina-a/Blender-models-thesis.git>.
The models are shown in Figure 3.15. The developed Python scripts enabling model exploration are also provided and can be used to query additional information associated with the gene expression patterns or analyze genes for coexpression.
- f) The instructions for running the programs, using the scripts and exploring the models are presented.

2. Files, programs, models and instructions for generating and analyzing the reconstructed *Platynereis* neuronal network are provided:

- a) The created model of the *Platynereis* neuronal network with integrated functionality for network exploration. The model can be also downloaded from GitHub: <https://github.com/albina-a/Blender-models-thesis.git> The model was shown in Figures 4.7 and 4.9.

- b) The developed Python scripts to create 3D models of neuronal reconstructions in Blender, to query additional information associated with the neuronal network and topological information, and to calculate network centrality metric. The functionality was demonstrated in Figure 4.5
 - c) Instructions for running the programs, using the scripts and exploring the model are presented.
3. Files, programs, models and instructions to map gene expression patterns obtained from the light microscopy (LM) to the electron microscopy (EM) datasets are provided:
- a) The developed programs and scripts to map the LM data to the EM data.
 - b) The affine and deformable transformations for mapping the LM and EM datasets.
 - c) Image stacks used as a reference to register LM and EM datasets (a modified DAPI average anatomy template and a modified EM image stack). The images are shown in Figure 5.4.
 - d) The original gene expression patterns and the gene expression patterns registered to the EM image stack. The registration results are shown in Figures 5.5 - 5.9.
 - e) Reconstructions of anatomical structures from the EM dataset used for the evaluation of the method. The images are presented in Figure 5.7 and 5.9.
4. The ImageJ/Fiji plugins, instructions and sample videos for the analysis of behavioral experiments in *Platynereis* larvae are provided:
- a) The ImageJ/Fiji plugin to measure phototaxis and a sample video.
 - b) The ImageJ/Fiji plugin to analyze swimming trajectories of ablated larvae and a sample video.
 - c) The ImageJ/Fiji plugin to classify and characterize larval locomotion, a sample video and a Weka model for classifying larvae behaviors.
 - d) Instructions on how to run the scripts and interpret the results.

The following files are enclosed in the dissertation and provided in digital form

1. Instructions for:

- a) rigid registration of light microscopy images
- b) affine and deformable registration of light microscopy images

-
- c) creating and using models of gene expression atlases in Blender
 - d) creating and using a Blender model of a neuronal network
 - e) registering light microscopy and electron microscopy data
 - f) analyzing of behavioral experiments in *Platynereis* larvae

2. Albina Asadulina, Aurora Panzera, Csaba Veraszto, Christian Liebig, Gaspar Jekely. Whole-body gene expression pattern registration in *Platynereis* larvae. *EvoDevo*, 3(1):27, 2012.

3. *In preparation*: Albina Asadulina, Markus Conzelmann, Elizabeth A. Williams, Aurora Panzera, Gáspár Jékely. Efficient representation of segmented light and electron microscopic volume data using Blender.

B. Instructions

Rigid registration of light microscopy images using ImageJ/Fiji

Orientation in Fiji.

The Fiji macro (Macro_rigid_fiji.ijm) orients the image of the 72 hpf larvae scanned in a random orientation. It should be possible to run the script in any version of Fiji.

- a) Open Macro_rigid_fiji.ijm in Fiji.
- b) Change the name of the input (dir) folder, that contains original OIF files.
- c) Change the name of the output (outdir) folder, that will contain oriented files.
- d) Specify template name (any oriented image or the template itself).
- e) Run the macro (ctrl+R). The script finds a transformation to orient images, applies it to all the channels and creates output in the output folder.
- f) Check the image. Larva in the output image should be oriented ventrally (ventral nerve cord should be in the first slices) with the head pointing upwards.

The macro version with TIFF files as input is Macro_rigid_fiji_tiff.ijm.

Orientation in ImageJ.

The following ImageJ plugins are necessary to run the programs: imagescience, TransformJ, Orientation.

1. Generate TIFF files:

- a) Open Macro_createTiff macro in ImageJ.
- b) Change the name of the input (dir) folder, that is an absolute path to the folder with original OIF files.
- c) Change the name of the output (outdir) folder, that will contain output files.
- d) Run the macro (ctrl+R). The script creates 3 TIFF files for each OIF file, ending with _dapi, _green, _red.

2a. Orienting images of 72 hpf larvae:

- a) Open Macro_orient72 in ImageJ.
- b) Change the name of the input (dir) folder, that contains TIFF files.
- c) Change the name of the output (outdir) folder, that will contain oriented files.
- d) Change matrix file name. It can be any file that you have permission to change. It will be

rewritten or created new.

e) Run the macro (ctrl+R). The script finds a transformation to orient image using dapi and tubulin channels, applies it to all the channels and creates output in the output folder.

f) Check the image. Larvae in the output image should be oriented ventrally with the head pointing upwards. If images are not oriented properly perform manual alignment (explained later).

2b. Orienting images of 48 hpf larvae:

a) Open Macro_orient48 in ImageJ.

b) Change the name of the input (dir) folder, that contains TIFF files.

c) Change the name of the output (outdir) folder, that will contain oriented files.

d) Change matrix file name. It can be any file that you have permission to change. It will be rewritten or created new.

e) Run the macro (ctrl+R). The script finds a transformation to orient image using dapi and tubulin channels, applies it to all the channels and creates output in the output folder.

f) Check the image. Larvae in the output image should be oriented ventrally with the head pointing upwards. If images are not oriented properly perform manual alignment (explained later).

3. Manual alignment:

a) Open TIFF file ending with _green in ImageJ (or Fiji).

b) Open 3D Viewer.

c) As soon as larva (axonal scaffold) appears in 3D Viewer, select it. When you select an object it has a square around it. Do not move it until you select it!

d) Orient axonal scaffold so that the nerve cord is in front (Animal's left is on the right side of the screen).

e) Go to menu Transformation/Save and save transform with the name of the original image file (without _green ending) in the folder containing TIFF files. Transformation will be save in *.mat file.

f) Open *.mat file and remove first 2 lines.

g) Open Macro_applyTransform in ImageJ.

h) Change the name of the input (dir) folder, that contains TIFF and *.mat files.

i) Change the name of the output (outdir) folder, that will contain oriented files.

j) Change values of standWidth and standHeight variables, that is the size of the output image. It should be 450x550 (450x600) for 72hpf and 500x500 for 48hpf.

k) Run the macro (ctrl+R). The script applies the transformation from the *.mat file to all the channels and creates output in the output folder.

Affine and deformable image registration

These are the instructions on building and running image registration project on a Linux operating system. Affine and deformable image registration are implemented using ITK toolkit. Input images should be roughly pre-aligned using rigid orientation.

1. Install ITK

You can find the installation instructions on the ITK webpage:

http://www.itk.org/Wiki/ITK/Getting_Started. The following installation guidelines succeeded with version 3.20 of ITK and 4.0. Later ITK versions might require some scripts modification.

a) Download ITK (from <http://www.itk.org/>) to `<source>` folder

b) Create `<bin>` folder

c) Open command line and go to the `<bin>` folder:

```
cd <bin>
```

d) Run the CMake:

```
cmake <source>
```

e) Build the project:

```
make
```

2. Build affine and deformable registration scripts

a) Create `<scripts>` folder

b) Copy all registration scripts into the `<scripts>` folder

c) Open command line

d) Define `ITK_DIR` variable:

```
export ITK_DIR=<bin>
```

e) Run the CMake:

```
cmake .
```

d) Build the project:

```
make
```

3. Register an individual image

Script `register_single.sh` registers a single sample image, applying registration scripts in the correct order:

- 'AffineRegistration' defines affine transform for registration using reference channel and applies corresponding affine transform to the gene expression channel.

- 'DeformableRegistration' defines deformable transform for registration using reference channel and applies corresponding deformable transform to the affined gene expression channel

To run the script you need to provide the following parameters in the correct order:

-
- 'fixedFileName' is a name of the file containing the template (8-bit image with the reference channel only)
 - 'movingFileName' is a name of the file containing a sample image that will be registered to the template (8-bit image with the reference channel only)
 - 'geneFileName' is the name of the file containing the gene expression channel of a sample image (8-bit image containing gene expression signal only)
 - 'outputFileMask' is mask for the output file (providing absolute path to the output file)

Example:

```
./register_single.sh template_dapi.tif sample_dapi.tif  
sample_red.tif /outputfolder/sample_registered
```

The register_single.sh script generates 7 output files:

Affine step:

- \langle outputFileMask \rangle .aff_transform.txt contains affine transform parameters
- \langle outputFileMask \rangle .aff.tiff contains sample image after affine transformation
- \langle outputFileMask \rangle .aff_red.tiff contains sample gene expression channel after affine transformation

Deformable step:

- \langle outputFileMask \rangle .def_field.mha contains deformable transform parameters
- \langle outputFileMask \rangle .def.tiff contains sample image after deformable transformation
- \langle outputFileMask \rangle .def_after.tiff contains overlaid template and sample images after deformable registration
- \langle outputFileMask \rangle .def_red.tiff contains sample gene expression channel after affine transformation. This is the result of the registration.

4. Register multiple images

Script register_folder.sh registers all images in a specified folder. The following parameters should be specified:

- 'imageFolder' is a folder containing sample images
- 'referenceImage' is a name of the file containing a template for registration
- 'outputFolder' is a folder, where output files will be stored.
- 'refChannel' is a string containing the ending of file containing reference channel (e.g. if you reference channel images end with _dapi.tif refChannel is 'dapi')

Example:

B. Instructions

```
./register_folder.sh /inputImages/ template_dapi.tif  
/outputFolder dapi
```

The script will register all images from the inputImages folder using dapi channel as reference signal.

5. Generate an average image

You can use the script 'NormalizeAndAverageFiles' to normalize (using mean and deviation of the image intensities) and average several images. The format is:

```
./NormalizeAndAverageFiles <input file list> <outputFile>
```

where <input file list> enumerates all input images separated with space and <outputFile> is the output file for an average image.

Script 'normAvgFiles.sh' normalizes and averages all files in a specified folder:

```
./normAvgFiles.sh <inputFolder> <outputFile>
```

Instructions for creating and using models of gene expression atlases in Blender

Exploring the gene expression atlases

1. Download Blender from: <http://www.blender.org/>
2. Open *.blend file in Blender.
3. Attribute queries: open and run (ctrl+p) the scripts Panel_genes.py, Panel_behavior.py or Panel_showGeneTypes.py in Blender to enable a panel for querying by gene name, gene functionality or gene type respectively. Panels will appear in the 'Tools' menu to set show/hide options for different groups. Press 'apply' to update changes.
4. Coexpression: open and run (ctrl+p) the script Panel_coexpression.py and run it to enable the gene coexpression queries. Select gene names in the dropboxes and press the 'Show coexpression' button to find coexpression between two genes. To remove the coexpression volumes from the project press 'Remove coexpressions'. To calculate coexpression matrix, start Blender from the terminal and press 'Coexpression matrix' in the 'Gene Coexpression' panel. The matrix will be printed out in the terminal. Copy it in Excel, format and save as a text file. Import as 'Text Image' in ImageJ and adjust the colors.

Creating a new model of a gene expression atlas

1. Create a new Blender project.
2. Import gene expression patterns in a surface form (we used 3D Viewer in Image/Fiji to generate surfaces). Adjust scale, location and materials. Python API can facilitate that.
3. To make a project compatible with queries, add gene expression domains to the 'Genes' group and Annotate with additional information if available. Gene functionality groups start with '_', gene types start with '-' and anatomical groups can start with anything else. This ensures that the groups appear in the corresponding panels.
4. To access the provided functionality, import the scripts Panel_genes.py, Panel_behavior.py or Panel_showGeneTypes.py in Blender, which will enable queries by gene name, gene functionality or gene type respectively.
5. Import/open the script Panel_coexpression.py and run it to enable gene coexpression queries and calculation of a coexpression matrix.

Instructions for creating and using models of a neuronal network in Blender

Exploring the neuronal network model

1. Download Blender from: <http://www.blender.org/> and open *.blend file in Blender.
2. Attribute queries: open and run scripts (ctrl+p) Panel_Behavior.py, Panel_Genes.py or Panel_Groups.py in Blender to enable queries for behaviour, expressed genes or anatomical groups respectively. Panels will appear in the 'Tools' menu to set show/hide options for different groups. Press 'apply' button to update the changes.
3. Connectivity queries: open and run (ctrl+p) script Panel_connectivity.py to enable connectivity queries. Select a cell body of a cell of interest to query for pre- or postsynaptic cells or subnetworks. Select cell bodies of two cells of interest and press 'show connectors' to highlight all synapses from the first selected cell to the second selected cell. Select attribute group names in the dropboxes and press 'show intergroup connections' to show all synapses between two groups.
4. Centrality: open and run (ctrl+p) script Panel_centrality.py and run it to enable the calculation of centrality metric. The type of centrality can be specified. Afterwards, colors can be reset to the original color-coding.

Creating a new neuronal network model

1. Create a new Blender project
2. To import neurons, open and run the script Panel_import_Catmaid.py or Panel_import_Trakem.py depending on your data source. A panel for import will appear in the 'Tools' menu. Specify OBJ file name (in the 'neuron file' field) for Trakem import or a neuroml file name (in the 'Catmaid file' field) for Catmaid import. After pressing the 'import' button, the neuronal structures will appear in Blender. Curate structures if necessary and adjust the materials.
3. To import synapses, specify a neuroml file for Catmaid import or two files for Trakem (connectivity graph and connector coordinates) in the same panel used for the import of neurons. After pressing 'import' synapses will appear in Blender.
4. To make project compatible with the attribute queries, add all neurons into the 'Cells' groups (you can use Python scripting here); add synapses into the 'connectors' group; add neurons into the 'Neurons' group; annotate with additional information if available: behavior groups start with '_', gene groups start with '-' and anatomical groups can start with anything else. This ensures that the groups appear in the corresponding panels.
5. To access the provided functionality, import scripts Panel_Behavior.py, Panel_Genes.py or Panel_Groups.py in Blender, which enable queries over behaviour, expressed genes or anatomical groups respectively.
6. Import script Panel_connectivity.py to enable connectivity queries.
7. Import/open script Panel_centrality.py and run it to enable the calculation of centrality metric.

Instructions for mapping gene expression patterns obtained from light microscopy to the image stack obtained from electron microscopy (EM)

The method finds the candidate cells in the EM dataset, where a gene is expressed. ITK toolkit and the provided scripts should be compiled in order to apply the transformations. The procedure is the same as for the gene expression atlas. ImageJ/Fiji should be installed. The program can run on Windows, OS X and Linux operating systems.

1. Rotate and remove image layers:

- a) Change image orientation from ventral to apical view using Reslice option ImageJ/Fiji.
- b) Remove the last layers starting from 286 using Slice Remover in ImageJ/Fiji.

2. Apply affine and deformable transforms to the image using the provided scripts. Run script rules in the terminal:

```
./testbash_apply_transform.sh fixedFileName movingFileName affineTransform deformTransform outputMask
```

where 'fixedFileName' is a EM image stack, 'movingFileName' is an image with a gene expression pattern from the step 1, 'affineTransform' is the provided affine transform, 'deformTransform' is the provided deformable transform and 'outputMask' is the output file mask including the full path and file-name prefix.

Example:

```
./testbash_apply_transform.sh EM_stack.tif FMRFa_rotated.tiff  
affine_transform.txt deformation_field.mha /output/FMRFa_registered
```

3. Output file will appear in the specified folder. The file ending with '_em_def_red.tiff' (e.g. 'FMRFa_registered_def_red.tiff') contains the transformed image mapped to the EM stack. Overlay the output image with the EM stack to find the candidate cells.

Instructions on how to use the ImageJ/Fiji plugin Phototoaxis_measure.

1. Open Phototoaxis_measure.java in ImageJ/Fiji.
2. Set variable 'imagePath' to the directory with image(s) and run the plugin.
3. The output will appear in the log window, indicating file name and average negative and average positive shift of larva positions and phototaxis metric value.

Instructions on the usage of the ImageJ/Fiji plugin Phototaxis_trajectory.

1. Open Phototaxis_trajectory.java in ImageJ/Fiji.
2. Open the video recording of the experiment and run the plugin.
3. The output will appear in the log window, indicating number of intersections and number of tracks.

Instructions on the usage of the ImageJ/Fiji plugin Locomotion_analysis.

1. Open Locomotion_analysis.java in ImageJ/Fiji.
2. Set variable 'videoFolderPath' to the directory with video recording(s).
3. Set variable 'outputPath' to the output directory.
4. Set variable 'modelPath' to the provided Weka model.
5. run the plugin.
6. The output will appear in the log window, indicating file name, swimming-to-crawling rate, average swimming speed, average crawling speed, as well as speed of the individual larvae.

Bibliography

- [1] D. A. Levitis, W. Z. Lidicker, and G. Freund. Behavioural biologists don't agree on what constitutes behaviour. *Animal Behaviour*, 78(1):103–110, Jul 2009.
- [2] N. Tinbergen. On aims and methods of ethology. *Zeitschrift fuer Tierpsychologie*, 20(4):410–433, 1963.
- [3] Adam Lampert, Alan Hastings, Edwin D. Grosholz, Sunny L. Jardine, and James N. Sanchirico. Optimal approaches for balancing invasive species eradication and endangered species management. *Science*, 344(6187):1028–1031, 2014.
- [4] Ravindra K. Sinha and Kurunthachalam Kannan. Ganges river dolphin: An overview of biology, ecology, and conservation status in India. *AMBIO*, pages 1–18, 2014.
- [5] D. Fischbein and J.C. Corley. Classical biological control of an invasive forest pest: a world perspective of the management of *Sirex noctilio* using the parasitoid *Ibalia leucospoides* (Hymenoptera: Ibalidae). *Bulletin of Entomological Research*, FirstView:1–12, 6 2014.
- [6] Laura K. Sirot, Rebecca L. Poulson, M. Caitlin McKenna, Hussein Girnary, Mariana F. Wolfner, and Laura C. Harrington. Identity and transfer of male reproductive gland proteins of the dengue vector mosquito, *Aedes aegypti*: Potential tools for control of female feeding and reproduction. *Insect Biochemistry and Molecular Biology*, 38(2):176 – 189, 2008.
- [7] Clelia F. Oliva, Maxime Jacquet, Jeremie Gilles, Guy Lemperiere, Pierre-Olivier Maquart, Serge Quilici, Francois Schooneman, Marc J. B. Vreysen, and Sebastien Boyer. The sterile insect technique for controlling populations of *Aedes albopictus* (Diptera: Culicidae) on reunion island: Mating vigour of sterilized males. *PLoS ONE*, 7(11):e49414, 11 2012.
- [8] Clelia F. Oliva, David Damiens, Marc J. B. Vreysen, Guy Lemperiere, and Jeremie Gilles. Reproductive strategies of *Aedes albopictus* (Diptera: Culicidae) and implications for the sterile insect technique. *PLoS ONE*, 8(11):e78884, 11 2013.

- [9] Rebecca A. Boulton, Laura A. Collins, and David M. Shuker. Beyond sex allocation: the role of mating systems in sexual selection in parasitoid wasps. *Biological Reviews*, 2014.
- [10] J. J. Skiff and D. A. Yee. Behavioral differences among four co-occurring species of container mosquito larvae: effects of depth and resource environments. *J. Med. Entomol.*, 51(2):375–381, Mar 2014.
- [11] Kevin L. Briggman, Moritz Helmstaedter, and Winfried Denk. Wiring specificity in the direction-selectivity circuit of the retina. *Nature*, 471(7337):183–188, March 2011.
- [12] N. Randel, A. Asadulina, L. A. Bezares-Calderon, C Veraszto, R. Shahidi, and G. Jekely. Neuronal connectome of a sensory-motor circuit for visual navigation. *Elife*, 2014.
- [13] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [14] M Calisti, M Giorelli, G Levy, B Mazzolai, B Hochner, C Laschi, and P Dario. An octopus-bioinspired solution to movement and manipulation for soft robots. *Bioinspiration and Biomimetics*, 6(3):036002, 2011.
- [15] Fei Li, Weiting Liu, Cesare Stefanini, Xin Fu, and Paolo Dario. A novel bioinspired PVDF micro/nano hair receptor for a robot sensing system. *Sensors*, 10(1):994–1011, 2010.
- [16] Nicholas A Swierczek, Andrew C Giles, Catharine H Rankin, and Rex A Kerr. High-throughput behavioral analysis in *C. elegans*. *Nature Methods*, 8(7):592–598, 2011.
- [17] Greg J. Stephens, Bethany Johnson-Kerner, William Bialek, and William S. Ryu. Dimensionality and dynamics in the behavior of *C. elegans*. *PLoS Comput Biol*, 4(4):e1000028, 04 2008.
- [18] Kuang-Man Huang, Pamela Cosman, and William R. Schafer. Machine vision based detection of omega bends and reversals in *C. elegans*. *Journal of Neuroscience Methods*, 158(2):323 – 336, 2006.
- [19] Joong-Hwan Baek, Pamela Cosman, Zhaoyang Feng, Jay Silver, and William R Schafer. Using machine vision to analyze and classify *Caenorhabditis elegans* behavioral phenotypes quantitatively. *Journal of Neuroscience Methods*, 118(1):9 – 21, 2002.
- [20] M. Kabra, A. A. Robie, M. Rivera-Alba, S. Branson, and K. Branson. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nature Methods*, 10(1):64–67, Jan 2013.

-
- [21] K. Branson, A. A. Robie, J. Bender, P. Perona, and M. H. Dickinson. High-throughput ethomics in large groups of *Drosophila*. *Nature Methods*, 6(6):451–457, Jun 2009.
- [22] H. Dankert, L. Wang, E. D. Hoopfer, D. J. Anderson, and P. Perona. Automated monitoring and analysis of social behavior in *Drosophila*. *Nat. Methods*, 6(4):297–303, Apr 2009.
- [23] L. Luo, M. Gershow, M. Rosenzweig, K. Kang, C. Fang-Yen, P. A. Garrity, and A. D. Samuel. Navigational decision making in *Drosophila* thermotaxis. *The Journal of Neuroscience*, 30(12):4261–4272, Mar 2010.
- [24] Jean-Rene Martin. A portrait of locomotor behaviour in *Drosophila* determined by a video-tracking paradigm. *Behavioural Processes*, 67(2):207 – 219, 2004.
- [25] Alice A. Robie, Andrew D. Straw, and Michael H. Dickinson. Object preference by walking fruit flies, *Drosophila melanogaster*, is mediated by vision and graviperception. *The Journal of Experimental Biology*, 213(14):2494–2506, 2010.
- [26] Alex Gomez-Marin, Nicolas Partoune, Greg J. Stephens, and Matthieu Louis. Automated tracking of animal posture and movement during exploration and sensory orientation behaviors. *PLoS ONE*, 7(8):e41642, 08 2012.
- [27] Andrew D. Straw, Kristin Branson, Titus R. Neumann, and Michael H. Dickinson. Multi-camera real-time three-dimensional tracking of multiple flying animals. *Journal of The Royal Society Interface*, 2010.
- [28] Reza Ardekani, Anurag Biyani, Justin E. Dalton, Julia B. Saltz, Michelle N. Arbeitman, John Tower, Sergey Nuzhdin, and Simon Tavare. Three-dimensional tracking and behaviour monitoring of multiple fruit flies. *Journal of The Royal Society Interface*, 2012.
- [29] F. de Chaumont, R. D. Coura, P. Serreau, A. Cressant, J. Chabout, S. Granon, and J. C. Olivo-Marin. Computerized video analysis of social interactions in mice. *Nat. Methods*, 9(4):410–417, Apr 2012.
- [30] Kuo-Hua Huang, Misha B. Ahrens, Timothy W. Dunn, and Florian Engert. Spinal projection neurons control turning behaviors in zebrafish. *Current Biology*, 23(16):1566–1573, 2013.
- [31] Satoru Kato, Takashi Nakagawa, Masato Ohkawa, Kenichiro Muramoto, Osamu Oyama, Akihito Watanabe, Hiroshi Nakashima, Tetsu Nemoto, and Kayo Sugitani. A computer image processing system for quantification of zebrafish behavior. *Journal of Neuroscience Methods*, 134(1):1 – 7, 2004.

- [32] Markus Conzelmann, Sarah-Lena Offenburger, Albina Asadulina, Timea Keller, Thomas A. Muench, and Gaspar Jekely. Neuropeptides regulate swimming depth of *Platynereis* larvae. *Proceedings of the National Academy of Sciences*, 108(46):E1174–E1183, 2011.
- [33] Markus Conzelmann, Elizabeth A. Williams, Sorin Tunaru, Nadine Randel, Reza Shahidi, Albina Asadulina, Juergen Berger, Stefan Offermanns, and Gaspar Jekely. Conserved mip receptor-ligand pair regulates *Platynereis* larval settlement. *Proceedings of the National Academy of Sciences*, 110(20):8224–8229, 2013.
- [34] Nadine Randel, Luis A. Bezares-Calderon, Martin Guehmann, Reza Shahidi, and Gaspar Jekely. Expression dynamics and protein localization of rhabdomeric opsins in *Platynereis* larvae. *Integrative and Comparative Biology*, 53(1):7–16, 2013.
- [35] Kit Yu Karen Chan, Houshuo Jiang, and Dianna K. Padilla. Swimming speed of larval snail does not correlate with size and ciliary beat frequency. *PLoS ONE*, 8(12):e82764, 12 2013.
- [36] Simon J. D. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, New York, NY, USA, 1st edition, 2012.
- [37] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007.
- [38] Mehmet Sezgin and Buelent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *J. Electronic Imaging*, 13(1):146–168, 2004.
- [39] Francesco Camastra, Alessandro Vinciarelli, and Jie Yu. Machine learning for audio, image and video analysis. *J. Electronic Imaging*, 18(2):029901, 2009.
- [40] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
- [41] J. M. Fitzpatrick and M. Sonka. "Handbook of Medical Imaging, Volume 2. Medical Image Processing and Analysis (SPIE Press Monograph Vol. PM80)". SPIE–The International Society for Optical Engineering, 1 edition, June 2000.
- [42] E.H.L Aarts and J Korst. *Simulated Annealing and Boltzmann Machines*. Wiley, Chichester, UK, 1989.

- [43] Juliane Zantke, Stephanie Bannister, Vinoth Babu Veedin Rajan, Florian Raible, and Kristin Tessmar-Raible. Genetic and genomic tools for the marine annelid *platynereis dumerilii*. *Genetics*, 197(1):19–31, 2014.
- [44] Alexei Fedorov and Larisa Fedorova. Where is the difference between the genomes of humans and annelids? *Genome Biology*, 7(1):203, 2006.
- [45] Albrecht Fischer and Adriaan Dorresteijn. The polychaete *Platynereis dumerilii* (Annelida): a laboratory animal with spiralian cleavage, lifelong segment proliferation and a mixed benthic/pelagic life cycle. *BioEssays*, 26(3):314–325, 2004.
- [46] Juliane Zantke, Tomoko Ishikawa-Fujiwara, Enrique Arboleda, Claudia Lohs, Katharina Schipany, Natalia Hallay, Andrew D. Straw, Takeshi Todo, and Kristin Tessmar-Raible. Circadian and circalunar clock interactions in a marine annelid. *Cell Reports*, 5(1):99–113, 2013.
- [47] Antje Fischer, Thorsten Henrich, and Detlev Arendt. The normal development of *Platynereis dumerilii* (nereididae, annelida). *Frontiers in Zoology*, 7(1):31, 2010.
- [48] Eve Gazave, Julien Behague, Lucie Laplane, Aurelien Guillou, Laetitia Preau, Adrien Demilly, Guillaume Balavoine, and Michel Vervoort. Posterior elongation in the annelid *Platynereis dumerilii* involves stem cells molecularly related to primordial germ cells. *Developmental Biology*, 382(1):246 – 267, 2013.
- [49] Markus Conzelmann, Elizabeth Williams, Karsten Krug, Mirita Franz-Wachtel, Boris Macek, and Gaspar Jekely. The neuropeptide complement of the marine annelid *Platynereis dumerilii*. *BMC Genomics*, 14(1):906, 2013.
- [50] Luis Ovidiu Popa, Oana Paula Popa, Ana-Maria Krapal, Elena Iulia Iorgu, and Victor Surugiu. Fine-scale population genetics analysis of *Platynereis dumerilii* (Polychaeta, Nereididae) in the black sea: How do local marine currents drive geographical differentiation? *Journal of Experimental Zoology Part A: Ecological Genetics and Physiology*, 321(1):41–47, 2014.
- [51] Mengting Yang and Xiangru Zhang. Comparative developmental toxicity of new aromatic halogenated dbps in a chlorinated saline sewage effluent to the marine polychaete *Platynereis dumerilii*. *Environmental Science and Technology*, 47(19):10868–10876, 2013.
- [52] Piero Calosi, Samuel P. S. Rastrick, Chiara Lombardi, Heidi J. de Guzman, Laura Davidson, Marlene Jahnke, Adriana Giangrande, Joerg D. Hardege, Anja Schulze, John I.

- Spicer, and Maria-Cristina Gambi. Adaptation and acclimatization to ocean acidification in marine ectotherms: an in situ transplant experiment with polychaetes at a shallow CO₂ vent system. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 368(1627), 2013.
- [53] Raju Tomer, Alexandru S. Denes, Kristin Tessmar-Raible, and Detlev Arendt. Profiling by image registration reveals common origin of annelid mushroom bodies and vertebrate pallium. *Cell*, 142(5):800–809, 2010.
- [54] Florian Raible, Kristin Tessmar-Raible, Kazutoyo Osoegawa, Patrick Wincker, Claire Jubin, Guillaume Balavoine, David Ferrier, Vladimir Benes, Pieter de Jong, Jean Weissenbach, Peer Bork, and Detlev Arendt. Vertebrate-type intron-rich genes in the marine annelid *Platynereis dumerilii*. *Science*, 310(5752):1325–1326, 2005.
- [55] Alexandru S. Denes, Gaspar Jekely, Patrick R.H. Steinmetz, Florian Raible, Heidi Snyman, Benjamin Prud'homme, David E.K. Ferrier, Guillaume Balavoine, and Detlev Arendt. Molecular architecture of annelid nerve cord supports common origin of nervous system centralization in bilateria. *Cell*, 129(2):277–288, 2007.
- [56] Greg W. Rouse. Trochophore concepts: ciliary bands and the evolution of larvae in spiralian Metazoa. *Biological Journal of the Linnean Society*, 66(4):411–464, 1999.
- [57] N. Rebscher, F. Zelada-Gonzalez, T.U. Banisch, F. Raible, and D. Arendt. Vasa unveils a common origin of germ cells and of somatic stem cells from the posterior growth zone in the polychaete *Platynereis dumerilii*. *Developmental Biology*, 306(2):599–611, 2007.
- [58] Pierre Kerner, Fabiola Zelada Gonzalez, Martine Gouar, Valerie Ledent, Detlev Arendt, and Michel Vervoort. The expression of a hunchback ortholog in the polychaete annelid *Platynereis dumerilii* suggests an ancestral role in mesoderm development and neurogenesis. *Development Genes and Evolution*, 216(12):821–828, 2006.
- [59] Milana Kulakova, Nadezhda Bakalenko, Elena Novikova, Charles E. Cook, Elena Eliseeva, Patrick R.H. Steinmetz, Roman P. Kostyuchenko, Archil Dondua, Detlev Arendt, Michael Akam, and Tatiana Andreeva. Hox gene expression in larval development of the polychaetes *Nereis virens* and *Platynereis dumerilii* (annelida, lophotrochozoa). *Development Genes and Evolution*, 217(1):39–54, 2007.
- [60] Jerome Hui, Florian Raible, Natalia Korchagina, Nicolas Dray, Sylvie Samain, Ghislaine Magdelenat, Claire Jubin, Beatrice Segurens, Guillaume Balavoine, Detlev Arendt, and David Ferrier. Features of the ancestral bilaterian inferred from *Platynereis dumerilii* parahox genes. *BMC Biology*, 7(1):43, 2009.

- [61] Renaud de Rosa, Benjamin Prud'homme, and Guillaume Balavoine. caudal and even-skipped in the annelid *Platynereis dumerilii* and the ancestry of posterior growth. *Evolution and Development*, 7(6):574–587, 2005.
- [62] Kristin Tessmar-Raible, Florian Raible, Foteini Christodoulou, Keren Guy, Martina Rembold, Harald Hausen, and Detlev Arendt. Conserved sensory-neurosecretory cell types in annelid and fish forebrain: Insights into hypothalamus evolution. *Cell*, 129(7):1389 – 1400, 2007.
- [63] A. N. Jha, T. H. Hutchinson, J. M. Mackay, B. M. Elliott, P. L. Pascoe, and D. R. Dixon. The chromosomes of *Platynereis dumerilii* (Polychaeta: Nereidae). *Journal of the Marine Biological Association of the United Kingdom*, 75:551–562, 8 1995.
- [64] O Simakov, TA Larsson, and D Arendt. Linking micro- and macro-evolution at the cell type level: a view from the lophotrochozoan *Platynereis dumerilii*. *Briefings in functional genomics*, 12(5):430–439, September 2013.
- [65] *Platynereis dumerilii* full-length normalized cDNA library. <http://jekely-lab.tuebingen.mpg.de/reagents/ESTs.html>.
- [66] D. Arendt, U. Technau, and J. Wittbrodt. Evolution of the bilaterian larval foregut. *Nature*, 409(6816):81–85, Jan 2001.
- [67] Adriaan W. C. Dorresteijn, Brian O'Grady, Albrecht Fischer, Eliane Porchet-Henneré, and Yolande Boilly-Marer. Molecular specification of cell lines in the embryo of *Platynereis* (Annelida). 202(5):260–269, May 1993.
- [68] Benjamin Backfisch, Vinoth Babu Veedin Rajan, Ruth M. Fischer, Claudia Lohs, Enrique Arboleda, Kristin Tessmar-Raible, and Florian Raible. Stable transgenesis in the marine annelid *platynereis dumerilii* sheds new light on photoreceptor evolution. *Proceedings of the National Academy of Sciences*, 110(1):193–198, 2013.
- [69] S. Bannister, O. Antonova, A. Polo, C. Lohs, N. Hallay, A. Valinciute, F. Raible, and K. Tessmar-Raible. TALENs mediate efficient and heritable mutation of endogenous genes in the marine annelid *Platynereis dumerilii*. *Genetics*, 197(1):77–89, May 2014.
- [70] C. Ackermann. *Markierung der Zelllinien im Embryo von Platynereis*. Ph.D. Thesis, University of Mainz, Mainz, New York, NY, USA, 2003.
- [71] Gaspar Jekely, Julien Colombelli, Harald Hausen, Keren Guy, Ernst Stelzer, Francois Nedelec, and Detlev Arendt. Mechanism of phototaxis in marine zooplankton. *Nature*, 456(7220):395–399, November 2008.

- [72] Vinoth Babu Veedin-Rajan, Ruth M. Fischer, Florian Raible, and Kristin Tessmar-Raible. Conditional and specific cell ablation in the marine annelid *Platynereis dumerilii*. *PLoS ONE*, 8(9):e75811, 09 2013.
- [73] Gaspar Jekely and Detlev Arendt. Cellular resolution expression profiling using confocal detection of nbt/bcip precipitate by reflection microscopy. *BioTechniques*, 42(6):751–755, Jun 2007.
- [74] Albina Asadulina, Aurora Panzera, Csaba Veraszto, Christian Liebig, and Gaspar Jekely. Whole-body gene expression pattern registration in *Platynereis* larvae. *EvoDevo*, 3(1):27, 2012.
- [75] AH Fischer and D Arendt. Mesoteloblast-like mesodermal stem cells in the polychaete annelid *Platynereis dumerilii* (Nereididae). *Journal of experimental zoology. Part B, Molecular and developmental evolution*, 320(2):94–104, March 2013.
- [76] Ann-Shyn Chiang, Chih-Yung Lin, Chao-Chun Chuang, Hsiu-Ming Chang, Chang-Huain Hsieh, Chang-Wei Yeh, Chi-Tin Shih, Jian-Jheng Wu, Guo-Tzau Wang, Yung-Chang Chen, et al. Three-dimensional reconstruction of brain-wide wiring networks in *Drosophila* at single-cell resolution. *Current Biology*, 21(1):1–11, 2011.
- [77] Fuhui Long, Hanchuan Peng, Xiao Liu, Stuart K Kim, and Eugene Myers. A 3d digital atlas of *C. elegans* and its application to single-cell analyses. *Nature methods*, 6(9):667–673, 2009.
- [78] Olaf Ronneberger, Kun Liu, Meta Rath, Dominik Ruess, Thomas Mueller, Henrik Skibbe, Benjamin Drayer, Thorsten Schmidt, Alida Filippi, Roland Nitschke, Thomas Brox, Hans Burkhardt, and Wolfgang Driever. ViBe-Z: a framework for 3d virtual colocalization analysis in zebrafish larval brains. *Nat. Methods*, 9(7):735–742, Jul 2012.
- [79] Ed S. Lein, Michael J. Hawrylycz, Nancy Ao, Mikael Ayres, Amy Bensinger, Amy Bernard, Andrew F. Boe, Mark S. Boguski, et al. Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, 445(7124):168–176, Jan 2007.
- [80] Arnim Jenett, Johannes E Schindelin, and Martin Heisenberg. The virtual insect brain protocol: creating and comparing standardized neuroanatomy. *BMC Bioinformatics*, 7:544, 2006.
- [81] Basil El Jundi, Wolf Huetteroth, Angela E. Kurylas, and Joachim Schachtner. Anisometric brain dimorphism revisited: Implementation of a volumetric 3D standard brain in *Manduca sexta*. *Journal of Comparative Neurology*, 517:210–225, 2009.

- [82] David Dreyer, Holger Vitt, Stefan Dippel, Brigitte Goetz, Basil el Jundi, Martin Kollmann, Wolf Huetteroth, and Joachim Schachtner. 3d standard brain of the red flour beetle *Tribolium castaneum*: a tool to study metamorphic development and adult plasticity. *Frontiers in Systems Neuroscience*, 4:3, 2010.
- [83] Michael D. Wong, Adrienne E. Dorr, Johnathon R. Walls, Jason P. Lerch, and R. Mark Henkelman. A novel 3d mouse embryo atlas based on micro-ct. *Development*, 139(17):3248–3256, 2012.
- [84] Benedicte Ballanger, Leon Tremblay, Veronique Sgambato-Faure, Maude Beaudoin-Gobert, Franck Lavenne, Didier Le Bars, and Nicolas Costes. A multi-atlas based method for automated anatomical *Macaca fascicularis* brain mri segmentation and pet kinetic extraction. *NeuroImage*, 77(0):26–43, 2013.
- [85] Emma Munoz-Moreno, Ariadna Arbat-Plana, Dafnis Batalle, Guadalupe Soria, Miriam Illa, Alberto Prats-Galino, Elisenda Eixarch, and Eduard Gratacos. A magnetic resonance image based atlas of the rabbit brain for automatic parcellation. *PLoS ONE*, 8(7):e67418, 2013.
- [86] Charless C. Fowlkes, Cris L. Luengo Hendriks, Soile V.E. Keraenen, Gunther H. Weber, Oliver Ruebel, Min-Yu Huang, Sohail Chatoor, Angela H. DePace, Lisa Simirenko, Clara Henriquez, Amy Beaton, Richard Weiszmann, Susan Celniker, Bernd Hamann, David W. Knowles, Mark D. Biggin, Michael B. Eisen, and Jitendra Malik. A quantitative spatiotemporal atlas of gene expression in the *Drosophila* blastoderm. *Cell*, 133(2):364–374, 2008.
- [87] C. Castro-Gonzalez, M.A. Luengo-Oroz, L. Douloquin, T. Savy, C. Melani, S. Desnoullez, M.J. Ledesma-Carbayo, P. Bourguine, N. Peyrieras, and A. Santos. Towards a digital model of zebrafish embryogenesis. integration of cell tracking and gene expression quantification. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 5520–5523, 2010.
- [88] Robert Brandt, Torsten Rohlfing, Juergen Rybak, Sabine Krofczik, Alexander Maye, Malte Westerhoff, Hans-Christian Hege, and Randolph Menzel. Three-dimensional average-shape atlas of the honeybee brain and its applications. *The Journal of Comparative Neurology*, 492(1):1–19, 2005.
- [89] Basil El Jundi, Stanley Heinze, Constanze Lenschow, Angela Kurylas, Torsten Rohlfing, and Uwe Homberg. The locust standard brain: a 3d standard of the central complex as a platform for neural network analysis. *Frontiers in Systems Neuroscience*, 3(21), 2010.

- [90] Ritobrato Datta, Jongho Lee, Jeffrey Duda, Brian B. Avants, Charles H. Vite, Ben Tseng, James C. Gee, Gustavo D. Aguirre, and Geoffrey K. Aguirre. A digital atlas of the dog brain. *PLoS ONE*, 7(12):e52140, 2012.
- [91] Pal Kvello, Bjarte B Lofaldli, Juergen Rybak, Randolph Menzel, and Hanna Mustaparta. Digital, three-dimensional average shaped atlas of the *Heliothis virescens* brain with integrated gustatory and olfactory neurons. *Frontiers in Systems Neuroscience*, 3(14), 2009.
- [92] Josien P. W. Pluim, J. B. Antoine Maintz, and Max A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003.
- [93] Ilwoo Lyu, Sun Hyung Kim, Joon-Kyung Seong, Sang Wook Yoo, Alan C. Evans, Yundi Shi, Mar Sanchez, Marc Niethammer, and Martin Styner. Cortical correspondence via sulcal curve-constrained spherical registration with application to macaque studies. volume 8669, Mar 2013.
- [94] B. T. Thomas Yeo, Mert R. Sabuncu, Tom Vercauteren, Nicholas Ayache, Bruce Fischl, , and Polina Golland. Spherical demons: Fast diffeomorphic landmark-free surface registration. *IEEE Transactions on Medical Imaging*, 29(3):650–668, 2010.
- [95] Hyunjin Park, Jun-Sung Park, Joon-Kyung Seong, Duk L. Na, and Jong-Min Lee. Cortical surface registration using spherical thin-plate spline with sulcal lines and mean curvature as features. *Journal of Neuroscience Methods*, 206(1):46 – 53, 2012.
- [96] Angela E. Kurylas, Torsten Rohlfing, Sabine Krofczik, Arnim Jenett, and Uwe Homberg. Standardized atlas of the brain of the desert locust, *Schistocerca gregaria*. *Cell and Tissue Research*, 333(1):125–145, 2008.
- [97] Jean Livet, Tamily A. Weissman, Hyuno Kang, Ryan W. Draft, Ju Lu, Robyn A. Benis, Joshua R. Sanes, and Jeff W. Lichtman. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. *Nature*, 450(7166):56–62, Nov 2007.
- [98] Malcolm E. Fisher, Allyson K. Clelland, Andrew Bain, Richard A. Baldock, Paula Murphy, Helen Downie, Cheryll Tickle, Duncan R. Davidson, and Richard A. Buckland. Integrating technologies for comparing 3D gene expression domains in the developing chick limb. *Developmental Biology*, 317(1):13–23, 2008.

-
- [99] Markus Conzelmann and Gaspar Jekely. Antibodies against conserved amidated neuropeptide epitopes enrich the comparative neurobiology toolbox. *EvoDevo*, 3(1):23, 2012.
- [100] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri. Nih image to imagej: 25 years of image analysis. *Nature Methods*, 9(7):671–675, 2012.
- [101] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Tobias Pietzsch Mark Longair, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Jean-Yves Tinevez Benjamin Schmid, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9(7):676–682, 2012.
- [102] J. P W Pluim, J.B.A. Maintz, and M.A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003.
- [103] Johannes Schindelin and Mark Longair. Rigid registration plugin. <http://132.187.25.13/home/?category=Download&page=RigidRegistration>.
- [104] Albina Asadulina. ChannelMerger plugin. http://imagejdocu.tudor.lu/doku.php?id=plugin:color:channel_merger_:start.
- [105] Android operating system. <http://www.android.com/>.
- [106] Android software development kit. <https://developer.android.com/sdk>.
- [107] libgdx game development framework. <http://libgdx.badlogicgames.com/>.
- [108] Fbx converter. <https://github.com/libgdx/fbx-conv>.
- [109] Javascript programming language. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- [110] Three.js, javascript-based library for 3d graphics. <http://threejs.org/>.
- [111] Three.js importer, blender add-on. <https://github.com/mrdoob/three.js/tree/master/utils/exporters/blender>.
- [112] WebGL. <http://www.khronos.org/webgl/>.
- [113] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

- [114] Craig Montell. *Drosophila* visual transduction. *Trends in Neurosciences*, 35(6):356–363, 2012.
- [115] Detlev Arendt. Evolution of eyes and photoreceptor cell types. *The International Journal of Developmental Biology*, 47(7-8):563–571, 2003.
- [116] Wormbase. Virtual worm project. <http://caltech.wormbase.org/virtualworm/>.
- [117] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of the sematode *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 314(1165):1–340, November 1986.
- [118] Travis A. Jarrell, Yi Wang, Adam E. Bloniarz, Christopher A. Brittin, Meng Xu, J. Nichol Thomson, Donna G. Albertson, David H. Hall, and Scott W. Emmons. The connectome of a decision-making neural network. *Science*, 337(6093):437–444, July 2012.
- [119] Daniel J. Bumbarger, Metta Riebesell, Christian Rödelsperger, and Ralf J. Sommer. System-wide rewiring underlies behavioral differences in predatory and bacterial-feeding nematodes. *Cell*, 152(1-2):109–119, January 2013.
- [120] Lav R. Varshney, Beth L. Chen, Eric Paniagua, David H. Hall, and Dmitri B. Chklovskii. Structural properties of the *Caenorhabditis elegans* neuronal network. *PLoS Comput Biol*, 7(2):e1001066+, February 2011.
- [121] Dmitri B. Chklovskii, Shiv Vitaladevuni, and Louis K. Scheffer. Semi-automated reconstruction of neural circuits using electron microscopy. *Current Opinion in Neurobiology*, 20(5):667–675, October 2010.
- [122] Yunkyung Sohn, Myung-Kyu Choi, Yong-Yeol Ahn, Junho Lee, and Jaeseung Jeong. Topological cluster analysis reveals the systemic organization of the *Caenorhabditis elegans* connectome. *PLoS Comput Biol*, 7(5), May 2011.
- [123] Albert Cardona, Stephan Saalfeld, Ignacio Arganda, Wayne Poreanu, Johannes Schindelin, and Volker Hartenstein. Identifying neuronal lineages of *Drosophila* by sequence analysis of axon tracts. *The Journal of Neuroscience*, 30(22):7538–7553, June 2010.
- [124] Shin-ya Takemura, Arjun Bharioke, Zhiyuan Lu, Aljoscha Nern, Shiv Vitaladevuni, Patricia K. Rivlin, William T. Katz, Donald J. Olbris, Stephen M. Plaza, Philip Winston, Ting Zhao, Jane A. Horne, Richard D. Fetter, Satoko Takemura, Katerina Blazek, Lei-Ann Chang, Omotara Ogundeyi, Mathew A. Saunders, Victor Shapiro, Christopher

- Sigmund, Gerald M. Rubin, Louis K. Scheffer, Ian A. Meinertzhagen, and Dmitri B. Chklovskii. A visual motion detection circuit suggested by *Drosophila* connectomics. *Nature*, 500(7461):175–181, August 2013.
- [125] Davi D. Bock, Wei-Chung A. Lee, Aaron M. Kerlin, Mark L. Andermann, Greg Hood, Arthur W. Wetzel, Sergey Yurgenson, Edward R. Soucy, Hyon S. Kim, and R. Clay Reid. Network anatomy and *in vivo* physiology of visual cortical neurons. *Nature*, 471(7337):177–182, March 2011.
- [126] Sen Song, Per J. Sjöström, Markus Reigl, Sacha Nelson, and Dmitri B. Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biol*, 3(3):e68+, March 2005.
- [127] Moritz Helmstaedter, Kevin L. Briggman, Srinivas C. Turaga, Viren Jain, H. Sebastian Seung, and Winfried Denk. Connectomic Reconstruction of the Inner Plexiform Layer in the Mouse Retina. *Nature*, 500(7461):168–174, August 2013.
- [128] James R. Kremer, David N. Mastronarde, and J. Richard McIntosh. Computer Visualization of Three-Dimensional Image Data Using IMOD. *Journal of Structural Biology*, 116(1):71–76, January 1996.
- [129] Albert Cardona, Stephan Saalfeld, Johannes Schindelin, Ignacio Arganda-Carreras, Stephan Preibisch, Mark Longair, Pavel Tomancak, Volker Hartenstein, and Rodney J. Douglas. TrakEM2 software for neural circuit reconstruction. *PLoS ONE*, 7(6):e38011+, June 2012.
- [130] John C. Fiala. Reconstruct: a free editor for serial section microscopy. *Journal of microscopy*, 218(Pt 1):52–61, April 2005.
- [131] Stephan Saalfeld, Albert Cardona, Volker Hartenstein, and Pavel Tomančák. CATMAID: collaborative annotation toolkit for massive amounts of image data. *Bioinformatics*, 25(15):1984–1986, August 2009.
- [132] J. Beyer, A. Al-Awami, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger. ConnectomeExplorer: query-guided visual analysis of large volumetric neuroscience data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2868–2877, Dec 2013.
- [133] EMMONS LAB. Reconstruction software: Program elegance. <http://worms.aecom.yu.edu/pages/Reconstruction%20Software.html>.

- [134] J. Morales, L. Alonso-Nanclares, J. R. Rodriguez, J. Defelipe, A. Rodriguez, and A. Merchan-Perez. Espina: a tool for the automated segmentation and counting of synapses in large stacks of electron microscopy images. *Frontiers in Neuroanatomy*, 5:18, 2011.
- [135] Eyewire, a game to map the brain. <http://eyewire.org/>.
- [136] M. Helmstaedter, K. L. Briggman, and W. Denk. High-accuracy neurite reconstruction for high-throughput neuroanatomy. *Nature Neuroscience*, 14(8):1081–1088, Aug 2011.
- [137] Won-Ki Jeong, Johanna Beyer, Markus Hadwiger, Rusty Blue, Charles Law, Amelio Vazquez, Clay Reid, Jeff Lichtman, and Hanspeter Pfister. Ssecret and neurotrace: Interactive visualization and analysis tools for large-scale neuroscience datasets. *IEEE Computer Graphics and Applications*, 30:58–70, 05/2010 2010.
- [138] Verena Kaynig, Amelio Vazquez-Reina, Seymour Knowles-Barley, Mike Roberts, Thouis R. Jones, Narayanan Kasthuri, Eric Miller, Jeff Lichtman, and Narayanan Kasthuri Hanspeter Pfister. Large-scale automatic reconstruction of neuronal processes from electron microscopy images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013.
- [139] J. Funke, B. Andres, F.A. Hamprecht, A. Cardona, and M. Cook. Efficient automatic 3d-reconstruction of branching neurons from em data. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1004–1011, June 2012.
- [140] J. R. Anderson, S. Mohammed, B. Grimm, B. W. Jones, P. Koshevoy, T. Tasdizen, R. Whitaker, and R. E. Marc. The Viking viewer for connectomics: scalable multi-user annotation and summarization of large volume data sets. *Journal of microscopy*, 241(1):13–28, January 2011.
- [141] Elizabeth Jurrus, Melissa Hardy, Tolga Tasdizen, P. Thomas Fletcher, Pavel Koshevoy, Chi-Bin Chien, Winfried Denk, and Ross Whitaker. Axon tracking in serial block-face scanning electron microscopy. *Medical Image Analysis*, 13(1):180–188, February 2009.
- [142] Srinivas C. Turaga, Joseph F. Murray, Viren Jain, Fabian Roth, Moritz Helmstaedter, Kevin Briggman, Winfried Denk, and H. Sebastian Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural computation*, 22(2):511–538, February 2010.
- [143] Viren Jain, Joseph F. Murray, Fabian Roth, Srinivas Turaga, Valentin Zhigulin, Kevin L. Briggman, Moritz N. Helmstaedter, Winfried Denk, and H. Sebastian Seung. Supervised

- learning of image restoration with convolutional networks. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.
- [144] Kannan U. Venkataraju, Antonio R. C. Paiva, Elizabeth Jurrus, and Tolga Tasdizen. Automatic markup of neural cell membranes using boosted decision stumps. In *ISBI'09: Proceedings of the Sixth IEEE international conference on Symposium on Biomedical Imaging*, pages 1039–1042. IEEE Press, 2009.
- [145] Björn Andres, Ullrich Köthe, Moritz Helmstaedter, Winfried Denk, and Fred Hamprecht. Segmentation of SBFSEM Volume Data of Neural Tissue by Hierarchical Classification. In *Pattern Recognition*, volume 5096, chapter 15, pages 142–152. 2008.
- [146] V. Kaynig, T. J. Fuchs, and J. M. Buhmann. Geometrical consistent 3D tracing of neuronal processes in ssTEM data. *Medical Image Computing and Computer-Assisted Intervention*, 13(Pt 2):209–216, 2010.
- [147] Verena Kaynig, Thomas Fuchs, and Joachim M. Buhmann. Neuron geometry extraction by perceptual grouping in ssTEM images. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2902–2909, June 2010.
- [148] Christoph Sommer, Christoph Straehle, Ullrich Koethe, and Fred A. Hamprecht. "ilastik: Interactive learning and segmentation toolkit". In *8th IEEE International Symposium on Biomedical Imaging (ISBI 2011)*, 2011.
- [149] Yuriy Mishchenko. Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs. *Journal of Neuroscience Methods*, 176(2):276–289, January 2009.
- [150] Seymour Knowles-Barley, Nancy J. Butcher, Ian A. Meinertzhagen, and J. Douglas Armstrong. Biologically inspired EM image alignment and neural reconstruction. *Bioinformatics*, 27(16):2216–2223, August 2011.
- [151] S. Navlakha, P. Ahammad, and E. W. Myers. Unsupervised segmentation of noisy electron microscopy images using salient watersheds and region merging. *BMC Bioinformatics*, 14:294, 2013.
- [152] Martin O'Reilly, Arnd Roth, Michael Häusser, and Lewis D. Griffin. A circle-based method for detection of neural fibre cross-sections in classically stained 2d electron micrographs. In *Microscopic Image Analysis with Applications in Biology (MIAAB), Sixth international workshop*, 2011.

- [153] A. Lucchi, K. Smith, R. Achanta, G. Knott, and P. Fua. Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks With Learned Shape Features. *Medical Imaging, IEEE Transactions on*, 31(2):474–486, February 2012.
- [154] C. N. Straehle, U. Kothe, G. Knott, and F. A. Hamprecht. Carving: scalable interactive segmentation of neural volume electron microscopy images. *Medical Image Computing and Computer-Assisted Intervention*, 14(Pt 1):653–660, 2011.
- [155] Bjoern Andres, Ullrich Koethe, Thorben Kroeger, Moritz Helmstaedter, Kevin L. Briggman, Winfried Denk, and Fred A. Hamprecht. 3D Segmentation of SBFSEM Images of Neuropil by a Graphical Model over Supervoxel Boundaries. *Medical Image Analysis*, December 2011.
- [156] D. Laptev, A. Vezhnevets, S. Dwivedi, and J. M. Buhmann. Anisotropic ssTEM image segmentation using dense correspondence across sections. *Medical Image Computing and Computer-Assisted Intervention*, 15(Pt 1):323–330, 2012.
- [157] *Multiphase Geometric Couplings for the Segmentation of Neural Processes*, 2009.
- [158] Albert Cardona. Towards semi-automatic reconstruction of neural circuits. *Neuroinformatics*, 11(1):31–33, 2013.
- [159] A. Cardona, S. Saalfeld, S. Preibisch, B. Schmid, A. Cheng, J. Pulokas, P. Tomančák, and V. Hartenstein. An integrated micro- and macroarchitectural analysis of the *Drosophila* brain by computer-assisted serial section electron microscopy. *PLoS Biology*, 8(10), 2010.
- [160] Simon G. Sprecher, Albert Cardona, and Volker Hartenstein. The *Drosophila* larval visual system: High-resolution analysis of a simple visual neuropil. *Developmental Biology*, 358(1):33 – 43, 2011.
- [161] Albert Cardona, Camilla Larsen, and Volker Hartenstein. Neuronal fiber tracts connecting the brain and ventral nerve cord of the early *Drosophila* larva. *The Journal of Comparative Neurology*, 515(4):427–440, 2009.
- [162] Amandine Bery, Albert Cardona, Pedro Martinez, and Volker Hartenstein. Structure of the central nervous system of a juvenile acoel, *Symsagittifera roscoffensis*. *Development genes and evolution*, 220(3-4):61–76, September 2010.
- [163] Autodesk maya, a 3d animation software. <http://www.autodesk.com/products/autodesk-maya>.

- [164] Blender Foundation. Blender, a 3d animation software. <http://www.blender.org/>.
- [165] Amira. www.amira.com/.
- [166] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. *International AAAI Conference on Weblogs and Social Media*, 2009.
- [167] graph visualization software graphviz. <http://www.graphviz.org/>.
- [168] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage*, 52(3):1059 – 1069, 2010. Computational Models of the Brain.
- [169] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, August 2008.
- [170] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal, Complex Systems*:1695, 2006.
- [171] graph-tool, python module for manipulation and statistical analysis of graphs. <http://graph-tool.skewed.de/>.
- [172] Gaspar Jekely. Evolution of phototaxis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1531):2795–2808, 2009.
- [173] Detlev Arendt, Kristin Tessmar, Maria-Ines M. de Campos-Baptista, Adriaan Dorresteyn, and Joachim Wittbrodt. Development of pigment-cup eyes in the polychaete *Platynereis dumerilii* and evolutionary conservation of larval eyes in Bilateria. *Development*, 129(5):1143–1154, March 2002.
- [174] N. Randel, L. A. Bezares-Calderon, M. Guhmann, R. Shahidi, and G. Jekely. Expression dynamics and protein localization of rhabdomeric opsins in *Platynereis* larvae. *Integrative and Comparative Biology*, 53(1):7–16, Jul 2013.
- [175] Imagej plugin 3d viewer. <http://3dviewer.neurofly.de/>.
- [176] Maarten van Steen. *Graph theory and complex networks : an introduction*. Maarten van Steen, Lexington, 2010.

- [177] Dieter R. Klopfenstein and Ronald D. Vale. The lipid binding pleckstrin homology domain in unc-104 kinesin is necessary for synaptic vesicle transport in *Caenorhabditis elegans*. *Molecular Biology of the Cell*, 15(8):3729–3739, 2004.
- [178] Java programming language. <http://www.java.com/>.
- [179] Dominique Debanne, Emilie Campanac, Andrzej Bialowas, Edmond Carlier, and Gisele Alcaraz. Axon physiology. *Physiological Reviews*, 91(2):555–602, 2011.
- [180] Neuromorph, blender plugin set. <http://cvlab.epfl.ch/NeuroMorph>.
- [181] M. Baitaluk, M. Sedova, A. Ray, and A. Gupta. BiologicalNetworks: visualization and analysis tool for systems biology. *Nucleic Acids Research*, 34(Web Server issue):W466–471, Jul 2006.
- [182] Nathalie Henry and Jean-daniel Fekete. MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):677–684, September 2006.
- [183] NeesJan Eck and Ludo Waltman. Vos: A new method for visualizing similarities between objects. In Reinhold Decker and Hans-J. Lenz, editors, *Advances in Data Analysis*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 299–306. Springer Berlin Heidelberg, 2007.
- [184] Michael E. Smoot, Keiichiro Ono, Johannes Ruscheinski, Peng-Liang L. Wang, and Trey Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics (Oxford, England)*, 27(3):431–432, February 2011.
- [185] Daniel A. Keim. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, January 2002.
- [186] Daniel Keim, Huamin Qu, and Kwan-Liu Ma. Big-data visualization. *Computer Graphics and Applications, IEEE*, 33(4):20–21, July 2013.
- [187] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages, 1996. Proceedings.*, pages 336–343, Sep 1996.
- [188] Lev Muchnik, Sen Pei, Lucas C. Parra, Saulo D. S. Reis, Jose S. Andrade, Shlomo Havlin, and Hernan A. Makse. Origins of power-law degree distribution in the heterogeneity of human activity in social networks, 2013.

-
- [189] Daniel MacLean, Nataliya Elina, Ericka R. Havecker, Susanne B. Heimstaedt, David J. Studholme, and David C. Baulcombe. Evidence for large complex networks of plant short silencing rnas. *PLoS ONE*, 5(3):e9901+, March 2010.
- [190] Michael Schmuker, Thomas Pfeil, and Martin P. Nawrot. A neuromorphic network for generic multivariate data classification. *Proceedings of the National Academy of Sciences*, 111(6), January 2014.
- [191] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. Bradford books : Psychology. MIT Press, 1986.
- [192] Joshua L. Morgan and Jeff W. Lichtman. Why not connectomics? *Nature Methods*, 10(6):494–500, May 2013.
- [193] Roman S. Polishchuk, Elena V. Polishchuk, Pierfrancesco Marra, Saverio Alberti, Roberto Buccione, Alberto Luini, and Alexander A. Mironov. Correlative light-electron microscopy reveals the tubular-saccular ultrastructure of carriers operating between Golgi apparatus and plasma membrane. *The Journal of Cell Biology*, 148(1):45–58, 2000.
- [194] R. B. Knowles, J. H. Sabry, M. E. Martone, T. J. Deerinck, M. H. Ellisman, G. J. Bassell, and K. S. Kosik. Translocation of RNA granules in living neurons. *The Journal of Neuroscience*, 16(24):7812–7820, Dec 1996.
- [195] Tarun M. Kapoor, Michael A. Lampson, Polla Hergert, Lisa Cameron, Daniela Cimini, E. D. Salmon, Bruce F. McEwen, and Alexey Khodjakov. Chromosomes can congress to the metaphase plate before biorientation. *Science*, 311(5759):388–391, 2006.
- [196] D. G. Booth, M. Takagi, L. Sanchez-Pulido, E. Petfalski, G. Vargiu, K. Samejima, N. Imamoto, C. P. Ponting, D. Tollervey, W. C. Earnshaw, and P. Vagnarelli. Ki-67 is a PP1-interacting protein that organises the mitotic chromosome periphery. *Elife*, 3:e01641, 2014.
- [197] P. Resa-Infante, R. Thieme, T. Ernst, P. C. Arck, H. Ittrich, R. Reimer, and G. Gabriel. Importin- $\alpha 7$ is required for enhanced influenza a virus replication in the alveolar epithelium and severe lung damage in mice. *Journal of Virology*, 88(14):8166–8179, Jul 2014.
- [198] P. Verkade. Moving EM: the Rapid Transfer System as a new tool for correlative light and electron microscopy and high throughput for high-pressure freezing. *Journal of Microscopy*, 230(2):317–328, 2008.

- [199] K.L. McDonald. A review of high-pressure freezing preparation techniques for correlative light and electron microscopy of the same cells and tissues. *Journal of Microscopy*, 235(3):273–281, 2009.
- [200] N. Jimenez, van Donselaar E.G., D.A.M. de Winter, K. Vocking, Verkleij A.J., and J.A. Post. Gridded Aclar: preparation methods and use for correlative light and electron microscopy of cell monolayers, by TEM and FIB-SEM. *Journal of Microscopy*, 237(2):208–220, 2010.
- [201] M.J. Stensrud, F.A. Chaudhry, T.B. Leergaard, J.G. Bjaalie, and V. Gundersen. Vesicular glutamate transporter-3 in the rodent brain: Vesicular colocalization with vesicular γ -aminobutyric acid transporter. *Journal of Comparative Neurology*, 521(13):3042–3056, 2013.
- [202] Hannah E. J. Armer, Giovanni Mariggi, Ken M. Y. Png, Christel Genoud, Alexander G. Monteith, Andrew J. Bushby, Holger Gerhardt, and Lucy M. Collinson. Imaging transient blood vessel fusion events in zebrafish by correlative volume electron microscopy. *PLoS ONE*, 4(11):e7716, 11 2009.
- [203] Paul A. Sims and Jeff D. Hardin. Fluorescence-integrated transmission electron microscopy images. 369:291–308, 2007.
- [204] P. Fabrowski, A. S. Necakov, S. Mumbauer, E. Loeser, A. Reversi, S. Streichan, J. A. Briggs, and S. De Renzis. Tubular endocytosis drives remodelling of the apical surface during epithelial morphogenesis in *Drosophila*. *Nature Communication*, 4:2244, 2013.
- [205] Susan J. Nixon, Richard I. Webb, Matthias Floetenmeyer, Nicole Schieber, Harriet P. Lo, and Robert G. Parton. A single method for cryofixation and correlative light, electron microscopy and tomography of zebrafish embryos. *Traffic*, 10(2):131–136, 2009.
- [206] J. N. Skepper. Immunocytochemical strategies for electron microscopy: choice or compromise. *Journal of Microscopy*, 199(1):1–36, 2000.
- [207] W. P. Faulk and G. M. Taylor. An immunocolloid method for the electron microscope. *Immunochemistry*, 8(11):1081–1083, Nov 1971.
- [208] Juergen Roth. The silver anniversary of gold: 25 years of the colloidal gold marker system for immunocytochemistry and histochemistry. *Histochemistry and Cell Biology*, 106(1):1–8, 1996.
- [209] Thomas Misje Mathiisen, Erlend Arnulf Nagelhus, Bahareh Jouleh, Reidun Torp, Didrik Solie Frydenlund, Maria-Niki Mylonakou, Mahmood Amiry-Moghaddam, Luciene Covolan, Jo Kristian Utvik, Bjorg Riber, Karen Marie Gujord, Jorunn Knutsen,

- Oivind Skare, Petter Laake, Svend Davanger, Finn-Mogens Haug, Eric Rinvik, and Ole Petter Ottersen. Postembedding immunogold cytochemistry of membrane molecules and amino acid transmitters in the central nervous system. In *Neuroanatomical Tract-Tracing 3*, pages 72–108. Springer US, 2006.
- [210] Vidar Gundersen, Alexander Talgy Holten, and Jon Storm-Mathisen. GABAergic synapses in hippocampus exocytose aspartate on to nmda receptors: quantitative immunogold evidence for co-transmission. *Molecular and Cellular Neuroscience*, 26(1):156 – 165, 2004.
- [211] Albert Cardona, Stephan Saalfeld, Stephan Preibisch, Benjamin Schmid, Anchi Cheng, Jim Pulokas, Pavel Tomancak, and Volker Hartenstein. An integrated micro- and macroarchitectural analysis of the *Drosophila* brain by computer-assisted serial section electron microscopy. *PLoS Biol*, 8(10):e1000502, 10 2010.
- [212] Bioimage informatics annual meeting. <http://www.bioimageinformatics.org/>
- [213] Nature issue. focus on bioimage informatics. <http://www.nature.com/nmeth/focus/bioimageinformatics/index.html/>.
- [214] Bioinformatics oxford journals. <http://bioinformatics.oxfordjournals.org/>.
- [215] Bmc bioinformatics. <http://www.biomedcentral.com/bmcbioinformatics/>.
- [216] Hanchuan Peng. Bioimage informatics: a new area of engineering biology. *Bioinformatics*, 24(17):1827–1836, 2008.
- [217] Jason R. Swedlow, Ilya G. Goldberg, and Kevin W. Eliceiri. Bioimage informatics for experimental biology*. *Annual Review of Biophysics*, 38(1):327–346, 2009.
- [218] Kevin W Eliceiri, Michael R Berthold, Ilya G Goldberg, Luis Ibanez, B S Manjunath, Maryann E Martone, Robert F Murphy, Hanchuan Peng, Anne L Plant, Badrinath Roysam, Nico Stuurman, Jason R Swedlow, Pavel Tomancak, and Anne E Carpenter. Biological imaging software tools. *Nat Meth*, 9(7):697–710, July 2012.
- [219] Anne E Carpenter, Lee Kametsky, and Kevin W Eliceiri. A call for bioimaging software usability. *Nature Methods*, 9(7):666–670, June 2012.