

Dissertation
submitted to the
Combined Faculties for the Natural Sciences and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Presented by

Raju Tomer

born in: Ghaziabad, India

Oral-examination:

Group Leader: Dr. Detlev Arendt

Refrees: Prof. Thomas Holstein
Dr. Eileen Furlong

**The evolution of mushroom body and
telencephalic cell types, studied by single cell
expression profiling of *Platynereis dumerilii* larvae**

PhD Student:

Raju Tomer

Group Leader:

Dr. Detlev Arendt

Referees:

Prof. Thomas Holstein

Dr. Eileen Furlong

Acknowledgement

I am very thankful to Dr. Detlev Arendt for his excellent supervision, for his support and encouragement to pursue new ideas, for his constant faith in my abilities and for critically reading this manuscript.

I would also like to thank the following people: Heidi Snyman for her help with the transcription factor screen; Diana Bryant and Heidi Snyman for their excellent job in maintaining the *Platynereis* culture; Fay Christodoulou for critically reading parts of this manuscript and for her very pleasant presence; Michael Wahlers (IT Services, EMBL) for his help in installing the evodevo server; Kristin Tessmar-Raible for her help with the basic techniques of developmental biology research; Mette Handberg-Thorsager for critically reading parts of this manuscript; and the entire Arendt lab for a very pleasant atmosphere. I would also thank Benjamin Backfisch and Nicola Kegel for working with me for their Diplom project.

I am very thankful to my thesis advisory committee for their critical feedback: Dr. Toby Gibson, Prof. Thomas Holstein and especially Dr. Eileen Furlong. I am also thankful to GeneCore, ALMF and Szilard library for help with sequencing, image processing software licenses and literature support. I am also very grateful to Zoonet (Marie Curie RTN) for funding.

I am very thankful to my friends Alexandru Denes and Rudolf Walczak for their support, encouragement and many wonderful moments outside the lab. I also thank Rudolf Walczak for translating the short summary of this manuscript into German.

And last but not least, special thanks to my parents and brothers for their support and encouragement.

Table of Contents

Short Summary	12
Zusammenfassung.....	14
1. Introduction	16
1.1 Introduction to Evodevo	18
1.2 Platynereis dumerilii as a model system for evo-devo research	21
1.3 Mushroom Body in insect brains	23
1.3.1 Mushroom Body structure and Function	23
1.3.2 Evolution of mushroom bodies	25
1.3.3 Comparison between mushroom bodies and vertebrate brains.....	27
1.3.4 Molecular fingerprint of insect mushroom bodies.....	28
1.4 Mushroom body like structure in Platynereis dumerilii brain	30
1.5 Vertebrate telencephalon patterning	31
1.5.1 Vertebrate telencephalon development.....	31
1.5.2 Telencephalon patterning and the emerging cell types.....	35
1.5.3 Evolution of telencephalon	37
1.6 Aim of the thesis	38
2. RESULTS	40
2.1 Whole Mount In Silico Expression Profiling.....	42
2.1.1 The WMISEP Protocol	42
2.1.1.1 The Basic idea.....	42
2.1.1.2 Overview of WMISEP	44
2.1.1.3 Registration of Platynereis larval brain images	46
2.1.1.4 Average reference brain image of 48 hour old Platynereis larvae.....	49
2.1.1.5 Proof of Principle experiments for WMISEP	51
2.1.1.6 Averaging expression patterns.....	53
2.1.1.7 Features extraction: Expression patterns to cellular expression profiles ..	55
2.1.1.8 Search interfaces for the database of cellular expression profiles	64
2.1.1.9 Extending WMISEP to other stages of development of Platynereis dumerilii.....	64
2.1.2 Clustering analysis	71
2.1.2.1 Clustering approaches	71
2.1.2.2 Clustering of Platynereis larval brain cells based on gene expression profile.....	73
2.1.2.3 Clustering of genes based on their spatial expression patterns.....	80
2.2 Evolution of Mushroom Body	83
2.2.1 Platynereis dumerilii Adult Brain anatomy	83
2.2.2 Platynereis Mushroom Body anatomy and Connections.....	88
2.2.3 Platynereis Mushroom Body Development.....	93

2.2.4 Platynereis Mushroom Body Molecular Fingerprint	98
2.2.5 Common origin of Insects and Platynereis Mushroom Body	100
2.2.6 Towards deciphering the function of Platynereis mushroom body	104
2.3 Evolution of vertebrate telencephalon	106
2.3.1 Vertebrate telencephalon regionalization genes are expressed in similar topological order in Platynereis dumerilii	106
2.3.3 Common origins of telencephalon and Mushroom Body specifying gene network?	108
3. DISCUSSION	110
3.1 Whole Mount In silico Expression Profiling for Evo-Devo research	112
3.1.1 WMISEP protocol	112
3.1.2 Clustering of larval brain cells and genes based on expression profiles and spatial patterns	113
3.1.3 Future perspectives for WMISEP	115
3.2 Evolution of mushroom body and telencephalic cell types	119
3.2.1 Evolution of insect and annelid mushroom bodies	119
3.2.2 Homology of mushroom body and dorsal telencephalon cell types	121
4. Materials and Methods	126
4.1 Experimental materials and methods	128
4.1.1 Platynereis dumerilii culture	128
4.1.2 cDNA templates for degenerate PCR and 3',5' RACE reactions	128
4.1.3 Cloning of Dachshund (Dach)	128
4.1.4 Cloning of Svp/COUPTF1	131
4.1.5 Preparation of RNA probes for WMISH	132
4.1.6 WMISH combined with immunostaining against acetylated tubulin	133
4.1.7 Double fluorescence WMISH	136
4.1.8 Bright field and Confocal microscopy	137
4.1.9 BrdU assay	138
4.1.10 Cryosections	139
4.1.11 Immunostaining on cryosections	139
4.2 Computational material and methods	140
4.2.1 Computer Hardware	140
4.2.2 Softwares	140
4.2.3 Transcription Factors screen	141
4.2.4 mfpBLAST and BrainExplorer development	142
4.2.5 Image processing algorithm development and testing	143
4.2.6 Clustering analysis	143
5. REFERENCES	144
6 APPENDIX	154
6.1 Source Codes of Programs	156
6.1.1 Image registration algorithms testing and development	156
6.1.2 Averaging expression patterns	161

6.1.3 Cellular modelling of Platynereis larval brain	162
6.1.4 Expression to model assignment program	168
6.1.5 mfpBLAST	171
6.1.6 PduBrainExplorer	177
6.2 Sequences and trees	179
6.2.1 Dach Sequence and tree	179
6.2.2 Svp/COUPTF1 Sequence and tree.....	181
6.2.3 Table of Platynereis Transcription Factors candidates identified in the screen	182

List of Figures

Figure 1. Using the phylogenetic tree of living animals to infer the direction of trait loss or gain in evolution.....	20
Figure 2. The life cycle of <i>Platynereis dumerilii</i>	21
Figure 3. The structural organization of mushroom body in Insects.....	23
Figure 4. Variations in the structure of mushroom bodies in Insects.....	25
Figure 5. The organization of the adult <i>Platynereis dumerilii</i> brain.....	30
Figure 6. Development of vertebrate brain.....	32
Figure 7. Dorso-ventral patterning of telencephalon.....	33
Figure 8. The cell types emerging from ventral telencephalon.....	35
Figure 9. Evolution of telencephalon.....	37
Figure 10. The basic idea of Whole Mount In Silico expression Profiling (WMISEP) protocol.....	42
Figure 11. Overall summary of Whole Mount In Silico Expression Profiling (WMISEP) Protocol.....	44
Figure 12. The general workflow of image registration algorithms.....	46
Figure 13. Procedure used to generate the average reference image of axonal scaffold of 48 hour old <i>Platynereis</i> larval brain.....	48
Figure 14. Three dimensional reconstruction of the average reference axonal scaffold of 48 hour old <i>Platynereis</i> larval brain.....	49
Figure 15. Proof of principle experiments for WMISEP.....	51
Figure 16. Averaging the images of spatial expression pattern acquired from multiple embryos.....	52
Figure 17. Generating cellular brain model of 48 hour old <i>Platynereis</i> larvae.....	55
Figure 18. Summary of the average expression patterns of all the genes investigated in 48 hour old larval brain.....	63
Figure 19. Interfaces for searching the database.....	63
Figure 20. Three-dimensional reconstruction of the average reference axonal scaffold of 56 hour old <i>Platynereis</i> larval brain.....	66
Figure 21. Proof of principle experiment of WMISEP protocol for 56 hpf.....	67
Figure 22. Collection of average expression patterns of genes investigated in 56 hour old larvae.....	68
Figure 23. Collection I of average expression patterns of genes investigated in 4 days old <i>Platynereis</i> Nechtochaete.....	69
Figure 24. Collection II of average expression patterns of genes investigated in 4 days old <i>Platynereis</i> Nechtochaete.....	70
Figure 25. Hierarchical clustering of <i>Platynereis</i> larval brain cells, based on gene expression profiles.....	72
Figure 26. Dendrogram representation of hopach clustering results of all the cells in 48 hour old <i>Platynereis</i> larval brain.....	76
Figure 27. Statistical significance and mediods of the three super clusters of larval brain cells as revealed by hopach analysis.....	77

Figure 28. Spatial localization of the larval brain cells belonging to the three super clusters revealed by hopach analysis.	78
Figure 29. Dendrogram representation of hopach clustering results of all <i>Dach</i> positive cells in 48 hour old <i>Platynereis</i> larval brain.	79
Figure 30. Hierarchical clustering of genes based on their spatial expression patterns. ..	80
Figure 31. Case studies of clusters of genes	81
Figure 32. Anatomy of Adult <i>Platynereis</i> brain.	85
Figure 33. Three dimensional reconstruction of brain of adult <i>Platynereis dumerilii</i>	85
Figure 34. Three dimensional model of the brain of adult <i>Platynereis dumerilii</i>	86
Figure 35. Serotonergic and FMRFamide positive neurons in relation to <i>Platynereis</i> mushroom bodies.	87
Figure 36. Tracing back the development of mushroom bodies anatomically.	90
Figure 37. <i>Dach</i> is expressed in adult <i>Platynereis</i> mushroom bodies and their anlagen..	91
Figure 38. <i>Pax6</i> is expressed in adult <i>Platynereis</i> mushroom bodies and their anlagen. .	92
Figure 39. <i>Pax6</i> and <i>Dach</i> are co-expressed in <i>Platynereis</i> mushroom bodies and their anlagen	93
Figure 40. Screen for genes expressed in <i>Platynereis</i> mushroom body anlagen.....	95
Figure 41. List for genes not (or marginally) expressed in <i>Platynereis</i> mushroom body anlagen.	96
Figure 42. Verification of mushroom body molecular fingerprint genes	97
Figure 43. Expression of BF1, Otx, Svp, VGluT and VAcHT in mushroom body in older stages.	98
Figure 44. Visualizing mushroom bodies in alive <i>Platynereis</i> worms	102
Figure 45. Ablation of <i>Platynereis</i> mushroom bodies using LASER cutter.....	103
Figure 46. Expression of vertebrate telencephalon regionalization genes in <i>Platynereis</i> 48 hour larvae.	106
Figure 47. Schematic comparison of early telencephalon patterning genes in <i>Platynereis</i> and vertebrates.	108
Figure 48. Direct comparison of average axonal scaffold of 48 hpf and 56 hpf <i>Platynereis</i> larval brains.....	117
Figure 49. Direct comparison of gene expression in 56 hpf and 48 hpf <i>Platynereis</i> larvae.	118
Figure 50. Schematic summary of evolution of mushroom body and telencephalon.....	124
Figure 51. The pipeline used for screening the Transcription Factors represented in <i>Platynereis</i> EST sequences.	141

Short Summary

One of the many ways to understand the evolution of elaborate organs such as the brain is to investigate the different cell types that constitute that organ. Cell types are defined by a unique combination of genes (molecular fingerprint) that specify the distinct morphological and physiological features that are characteristic of that cell type. In order to study the cell types in the brain of the developing annelid *Platynereis dumerilii* I have investigated the co-expression of several genes at cellular resolution. For this, I have developed a protocol, the so-called Whole Mount In Silico Expression Profiling (WMISEP), utilizing advanced image processing algorithms, whole mount in situ hybridization, immunostaining against acetylated tubulin and whole mount reflection confocal microscopy. The basic idea of the protocol is to acquire two color confocal image stacks, with one channel containing expression information for gene and the other channel containing the information of the axonal scaffold. The information in the axonal scaffold channel is then used to align several such images to a common reference average axonal scaffold image, and thus bringing the expression patterns into the same coordinate system. I conducted several experiments to illustrate the cellular resolution sensitivity and specificity of the protocol. WMISEP has been used to generate cell resolution expression of 72 genes. I also developed a cellular model of the 48 hour old *Platynereis* larval brain, which facilitated the generation of cellular gene expression profiles. Subsequently, I used several clustering techniques to cluster the larval brain cells and genes based on their expression profiles and spatial patterns respectively.

As an example application of WMISEP, I investigated the evolution of mushroom bodies (MBs) and telencephalic cell types. Firstly, I investigated the anatomy, development and molecular fingerprint of *Platynereis* MB cells. Subsequently, I compared the anatomy and molecular fingerprint of *Platynereis* and insect's MBs to test for deep homology. Furthermore, I investigated the expression of early telencephalon regionalization genes in *Platynereis* and showed that the vertebrate telencephalon patterning genes are expressed

in a similar spatial orientation in the *Platynereis* larval brain, suggesting that the telencephalon patterning gene network already existed in the last common ancestor of all bilaterian animals. Finally, the *Platynereis* MB and vertebrate cortex/hippocampus develop from the same molecular regions with respect to the conserved molecular topography.

Zusammenfassung

Eine der zahlreichen Möglichkeiten, ein besseres Verständnis für die Evolution eines so komplizierten Organs wie des Gehirns zu gewinnen, besteht darin, die unterschiedlichen Zelltypen zu untersuchen, aus denen das Organ besteht. Verschiedene Zelltypen zeichnen sich durch einzigartige Kombinationen von ihnen exprimierter Gene aus (molekularer Fingerabdruck), welche die für sie besonderen morphologischen und physiologischen Eigenschaften festlegen. Um die Zelltypen im Gehirn des sich entwickelnden Anneliden *Platynereis dumerilii* zu studieren, habe ich die Co-Expression zahlreicher Gene mit zellulärer Auflösung untersucht. Zu diesem Zweck habe ich eine Methode namens „Whole Mount In Silico Expression Profiling“ (WMISEP) entwickelt, welche auf der Verwendung hochentwickelter Bildverarbeitungsalgorithmen, *in situ* Hybridisierung ganzer Embryonen, Immunfärbung gegen acetyliertes Tubulin und „Whole mount Reflection“ Konfokalmikroskopie basiert. Das Konzept der Methode besteht darin, mit zwei Farben konfokale Bildstapel aufzunehmen, wobei ein Kanal die Information über die Expression des untersuchten Gens enthält, während im zweiten Kanal die markierten Axone als Referenz detektiert werden. Die axonalen Daten werden dann zu einem gemeinsamen axonalen Durchschnittsbild gemittelt, wodurch die Expressionsmuster der einzelnen Gene in ein einheitliches Koordinatensystem überführt werden. Ich habe eine Reihe von Experimenten durchgeführt, um die zelluläre Auflösung und hohe Spezifität der Methode unter Beweis zu stellen. Insgesamt wurde mit Hilfe von WMISEP die Expression von 72 Genen mit zellulärer Auflösung kartiert. Ich habe ein zelluläres Modell des Gehirns der 48h alten *Platynereis* Larve entwickelt, was mir die Herstellung zellulärer Genexpressionsprofile ermöglichte. Auf diesen aufbauend konnte ich mit Hilfe unterschiedlicher clustering Techniken die Zellen und Gene des larvalen Gehirns gemäß ihrer Expressionsprofile und räumlichen Muster gruppieren. Als beispielhafte Anwendung von WMISEP habe ich die Evolution der Pilzkörper (mushroom bodies) und telencephaler Zelltypen untersucht. Dafür wurden zunächst Anatomie, Entwicklung und molekularer Fingerabdruck von Pilzkörperzellen von *Platynereis* charakterisiert. Danach

habe ich diese Daten mit Anatomie und molekularem Fingerabdruck der Pilzkörper in Insekten verglichen, um zu klären, ob eine tiefergehende Homologie zwischen diesen Strukturen besteht. Zudem habe ich die Expression der für die frühe Ausbildung des Telencephalons verantwortlichen Gene in *Platynereis* untersucht und konnte nachweisen, dass ihre räumliche Anordnung derjenigen der entsprechenden Gene in Vertebraten ähneln. Diese Ergebnisse deuten darauf hin, dass das für die Ausbildung des Telencephalon verantwortliche Gennetzwerk bereits im letzten gemeinsamen Vorfahren aller Bilateria existierte. Zudem konnte ich nachweisen, dass sich die Pilzkörper von *Platynereis* und der Wirbeltiercortex/-hippocampus aus Regionen entwickeln, die über eine konservierte molekulare Topographie verfügen.

1. Introduction

1.1 Introduction to Evodevo

Evo-Devo is a relatively new field of biology. “*Evolutionary developmental biology (evo–devo) is the study of how developmental processes evolve to produce new patterns of development, new developmental gene regulation, new morphologies, new life histories and new behavioral capabilities*” (Raff, 2000). Historically, evo-devo arose from the combination of many different disciplines including comparative embryology, morphology, developmental biology, genetics, evolutionary theory and paleontology. Most of the current research in evo–devo aim to describe the last common ancestor (Urbilateria) of protostomes and deuterostomes, the major superphyla of bilaterian animals, by comparing the morphologies, developmental processes, developmental gene networks, cell types and genomes of bilaterian animals. The foremost requirement in evo-devo research is to have an accurate phylogenetic tree of the animal kingdom, which is essential for mapping the direction of gain and loss of a trait (e.g. gene, cell type). Was the trait present in the last common ancestor and lost in some lineages? Or was it independently acquired in some lineages (Figure 1B)?

Some of the first attempts for constructing the phylogenetic relationships among animals were based on the morphological comparisons. For instance, (Holmgren, 1916) made one of the first attempts at constructing the phylogenetic tree of Arthropoda based on their brain anatomy. Though morphological characteristics (e.g. brain anatomy) provide a good basis for understanding the relationships among closely related animals, they can be misleading when comparing very diverse group of animals. With the advent of DNA sequencing techniques and robust tree making algorithms, better trees are being inferred based on the genetic information. Moreover, recent advances in sequencing technologies is providing a wealth of genomic sequence information for further refining and in some cases drastically changing (Adoutte et al., 2000; Rokas and Carroll, 2006) the topology of the tree of life. The current understanding of the phylogenetic relationships among bilaterian animals is summarized in a simplified schematic tree in Figure 1A. Briefly, the last common ancestor of all bilaterian animals, Urbilateria, is thought to have existed more than 600 million years ago. Further in evolution, Urbilateria gave rise to two major superphyla – Protostomes and Deuterostomes. Subsequently, Protostomes further gave

rise to Ecdysozoa and Lophotrochozoan. The ecdysozoan superphylum includes *Drosophila*, honeybee, cockroaches, *C. elegans* and others; the lophotrochozoan superphylum includes earthworms, *Octopus*, *Platynereis dumerilii* and others; and Deuterostome superphylum includes human, mouse, fish, sea Urchins, *Ciona*, and amphioxus among others.

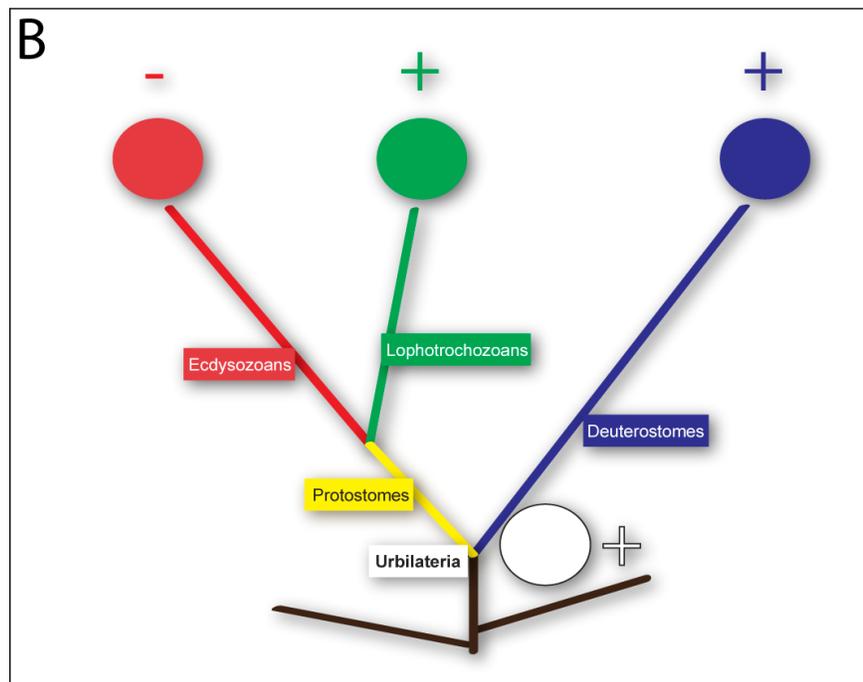
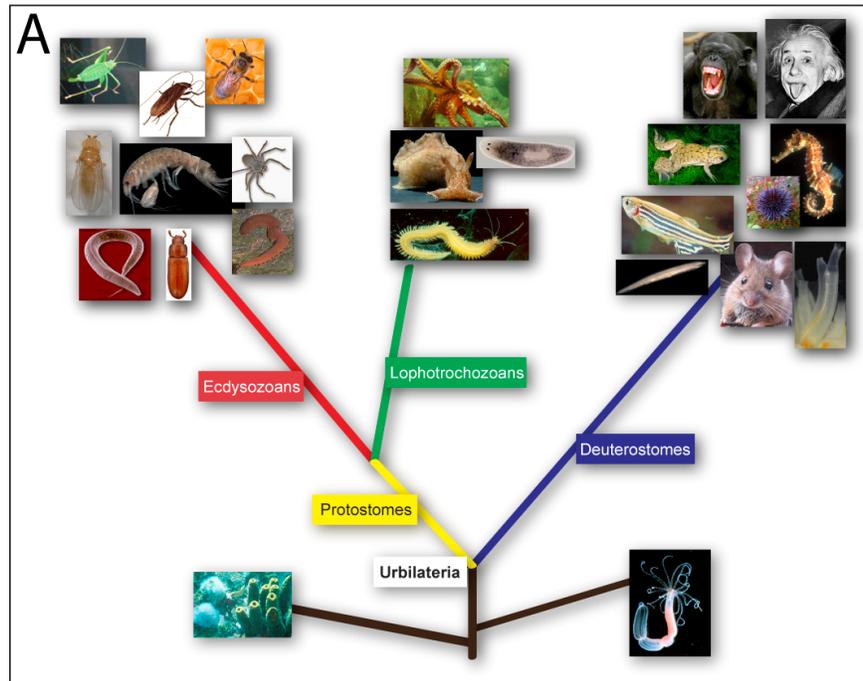


Figure 1. Using the phylogenetic tree of living animals to infer the direction of trait loss or gain in evolution

(A) The three major super-phyyla of bilaterian animals. Blue marks the lineage of Deuterostome super-phyllum and Yellow marks the Protostome lineage, which further split into Ecdysozoa (Red) and Lophotrochozoa (Green) superphyyla. The last common ancestor of all bilaterian animals, Urbilateria, is shown in white box. (B) a hypothetical scenario of a trait (which could be a cell type or a gene) not found in Ecdysozoa but found in Lophotrochozoan and Deuterostome. This would imply that this trait was present in the last common ancestor (Urbilateria) and must be secondarily lost in Ecdysozoa.

Using the sequenced genomes and EST libraries of many distantly related species, several comparative studies of the gene content and functions across the animal kingdom have shown that most of the genetic toolkit was already present in Urbilateria (Carroll, 2000). This observation gave rise to an obvious question: How can a conserved genetic toolkit generate such an enormous diversity of living life forms? The attempts to explain this paradox suggested that the phenotypic diversity originates from the differences in the spatial and temporal expression of genes rather than the products encoded by the genes (Jacob, 1977; King and Wilson, 1975).

Motivated by the discovery of Hox genes in *Drosophila* and the similarities of their expression patterns in many diverse species (McGinnis et al., 1984), most of the earlier research in evo-devo focused on comparing the spatial expression patterns of the developmental genes across species, with functional studies in relatively few model species (mainly *Drosophila*, *C. elegans* and mouse). Recently, much work has been done in understanding the evolution of developmental gene regulatory networks (Davidson and Erwin, 2006) and cell types (Arendt, 2008). One of the many ways to understand the evolution of elaborate organs such as the brain is to investigate the different cell types that constitute that organ. Cell types are defined by a unique combination of genes (molecular fingerprint, (Arendt, 2008)) that specify the distinct morphological and physiological features that are characteristic of that cell type. For instance, (Tessmar-Raible et al., 2007), by comparing the molecular fingerprint, showed that vertebrate hypothalamus-like cell types are found in marine annelids, suggesting that such cell types must have already existed in Urbilateria.

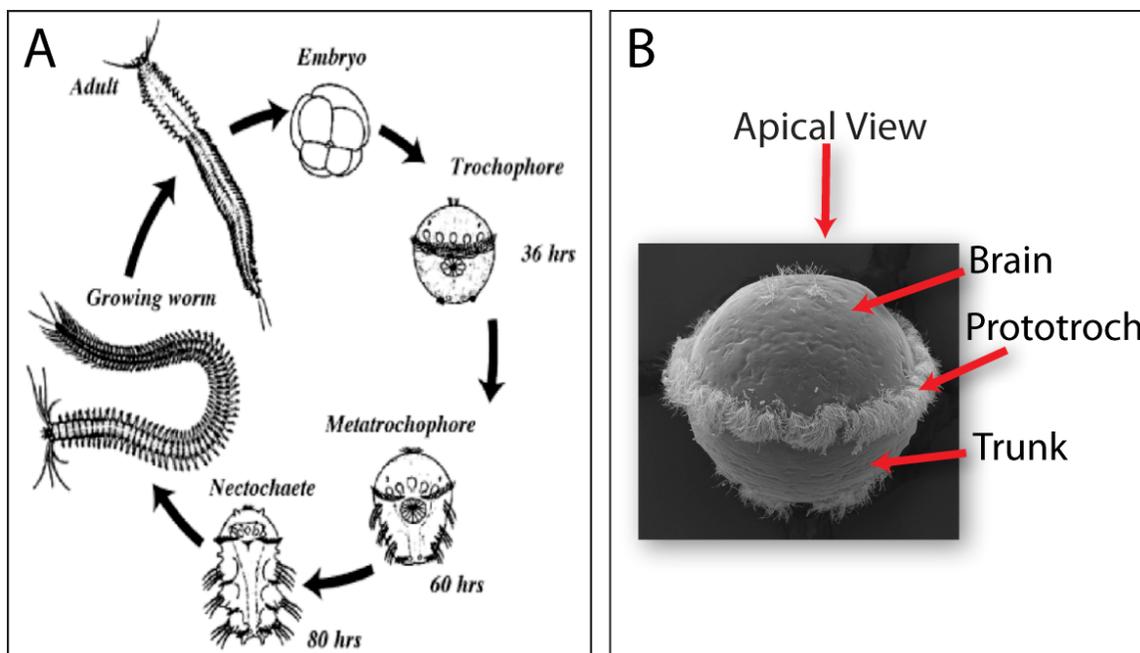


Figure 2. The life cycle of *Platynereis dumerilii*.

(A) *Platynereis dumerilii* has external fertilization. The mature male and female worms spawn by releasing eggs and sperm into the sea water where fertilization takes place. The developing embryos undergo spiral cleavages to hatch as planktonic trochophore larvae, which start to swim towards light. After about 80 hours of development, the larvae are transformed into three segmented Nectochaete and settle down in the benthic environment. The life span of *Platynereis dumerilii* varies from 3 to 18 months, at the end of which they undergo metamorphosis to become sexually mature. (B) The SEM (scanning electron microscopy) picture of 48 hour trochophore larva. The ciliated prototroch divides the embryo into brain and trunk. Most of the pictures in this thesis are presented from apical views, as shown by an arrow. Images courtesy of Guillaume Balavoine and Harald Hausen.

1.2 *Platynereis dumerilii* as a model system for evo-devo research

Platynereis dumerilii is a marine annelid belonging to the lophotrochozoan superphylum. Figure 2A describes the life cycle of *Platynereis dumerilii*. The mature “epitokous” worms go for excursion in the sea to find partners, where they spawn by releasing eggs and sperm in the sea water (Ackermann et al., 2005). Upon fertilization the eggs produce a jelly coat to stay afloat and to avoid other sperm cells. After about 18 hours of development, the eggs hatch as planktonic trochophore larvae and start to swim towards light. Subsequently, the larval and post-larval development takes place up to 3 days after fertilization. After about 80 hours post fertilization, the larvae are transformed

into three-segmented nectochaete larvae, which settle down in the benthic environment. These young worms grow by adding new segments posteriorly throughout their life and after about 3 to 18 months, they undergo metamorphosis to become sexually mature epitoke and enter a brief pelagic excursion life style to find partners.

The 48 hours old larvae already possess an impressive brain, containing approximately 2000 cells. As shown in Figure 2B, the brain and the trunk of 48 hour old *Platynereis* larvae are separated by a ciliated belt, Prototroch, which they use for swimming towards light. The larval development of *Platynereis dumerilii* is highly synchronized and stereotypic, and therefore it leads to the availability of several hundred larvae of identical developmental stage for experimental manipulation. Another advantage of *Platynereis* larvae is that they are completely transparent and hence ideal for imaging techniques. Furthermore, several studies conducted in our laboratory and elsewhere have shown that *Platynereis dumerilii* possess ancestral morphological features (Prud'homme et al., 2003; Raible and Arendt, 2004; Raible et al., 2005; Tessmar-Raible and Arendt, 2003), gene repertoire (Raible et al., 2005) and cell types (Arendt et al., 2008; Denes et al., 2007; Tessmar-Raible et al., 2007). Moreover, it belongs to the lophotrochozoan super phylum, which is scarcely represented in the model species used for the evo-devo research. In addition, the availability of several EST libraries, BAC libraries and many experimental manipulation techniques make *Platynereis dumerilii* an ideal model system for the evo-devo research.

1.3 Mushroom Body in insect brains

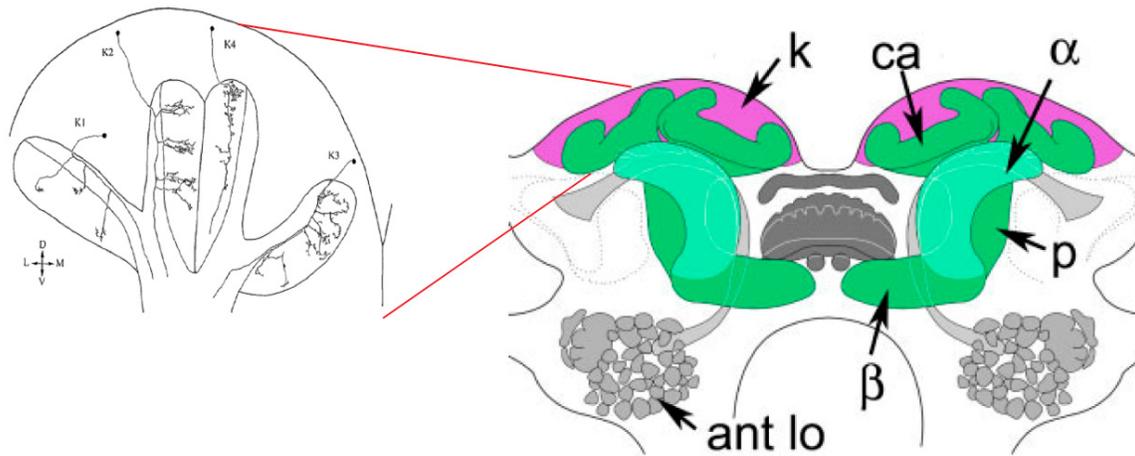


Figure 3. The structural organization of mushroom body in Insects.

The schematic describes the structure of mushroom body in cockroach's brain. The mushroom body is composed of highly clustered globuli cells (pink), called Kenyon cells (k), which supply parallel axonal projections to give rise to pedunculus (p) that further bifurcate to result in medial (β) and vertical (α) lobes. The dendrites of Kenyon cells cluster together to make cup-shaped structures, so called calyces (ca). In cockroaches mushroom bodies receive olfactory input connections from antennal lobes (ant lo). k: Kenyon Cells, ca: calyces, p: pedunculus, α : vertical lobes, β : medial lobes. The figure is adapted from (Brown and Strausfeld, 2006; Mizunami et al., 1998a)

1.3.1 Mushroom Body structure and Function

Mushroom Bodies (MBs) are prominent lobed neuropils, made of parallel axons bundle which are supplied by clusters of small-diameter globuli cells, located dorsally in the anterior part of the brain. Mushroom bodies are found in most of the arthropod groups (except crustaceans) and many marine annelids (e.g. *Platynereis dumerilii*, nereidid worms, scale worms). They were first reported in 1850 by Dujardin (Dujardin, 1850), who called them “corps pédonculés”. The first formal definition of mushroom bodies was given by Flögen (Flögel, 1876; Strausfeld et al., 1998) as: “the presence in the supraoesophageal mass of paired groups of several hundred to several hundred thousand minute cells (now called Kenyon cells) that surmount lobed neuropils”. Figure 3 shows the schematic view of the cockroach's mushroom bodies (Brown and Strausfeld, 2006; Mizunami et al., 1998a). The main structural features of mushroom bodies are labeled in the scheme - clustered globuli cells (Kenyon cells), parallel axons bundle forming the

pedunculus, cup shaped calyces made by dendrites of Kenyon cells, and medial and dorsal lobes resulting from bifurcations of the peduncle. In general, calyces receive sensory input and lobes represent the primary output. Mushroom bodies receive four different kinds of inputs: afferents to the mushroom body lobes, afferents to the calyces from protocerebrum and deutocerebrum carrying multimodal information, afferents from the antennal glomeruli and afferents from optic lobes of some Hymenoptera (e.g. Honey Bee, Ants).

Since their discovery, mushroom bodies have been functionally linked with intelligent behavior. One of the first indication about the function of mushroom bodies came from the comparative work in social Hymenoptera (e.g. Honey Bee) (Alten, 1910; Forel, 1874; Jonescu, 1909), where it was observed that queens and workers possessed bigger mushroom bodies compared to drones, suggesting the correlation between the size of mushroom bodies and the range of behaviors shown by the animal. Furthermore, relatively recent reports have implicated mushroom bodies in learning and memory. First indication came from the lesion experiments in ants where it was reported that the ants with perturbed mushroom bodies lost their ability to negotiate a maze using olfactory cues (Vowles, 1964). Furthermore, ablation experiments in cockroaches have shown the involvement of mushroom bodies in place memory formation (Mizunami et al., 1998b). The idea that mushroom bodies are the actual seat of memory and learning was corroborated by detailed investigations in *Drosophila* (de Belle and Heisenberg, 1994; Han et al., 1992; Heisenberg et al., 1985), where it was shown that mutant flies, having defective mushroom bodies, showed compromised olfactory memory and learning abilities. Additionally, the mushroom bodies have also been implicated in controlling sleep-like behavior in *Drosophila*. (Joiner et al., 2006; Koh et al., 2006; Pitman et al., 2006; Yuan et al., 2006).

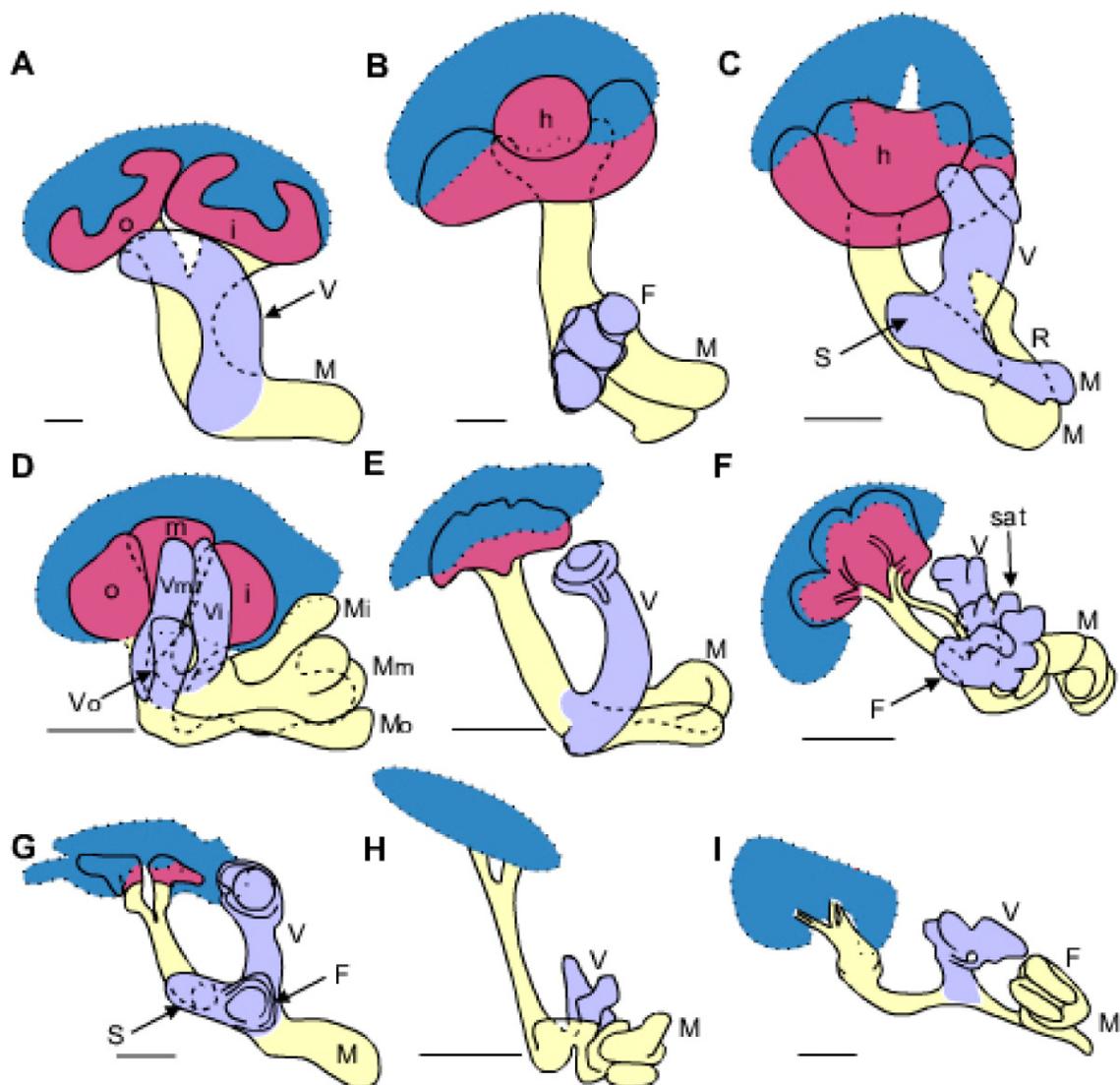


Figure 4. Variations in the structure of mushroom bodies in Insects.

The figure is based on (Strausfeld et al., 1998). The schema illustrates the variations in the organization of clustered Kenyon cells (blue), calyces (red), pediculi and lobes (light yellow) in odor sensitive (A-F) and odor insensitive (G-I) insects species. (A) Cockroach, (B) *Barytettix psolus*, (C) Cricket, (D) *Labidura riparia*, (E) caterpillar hunter beetle, (F) *Huebneriana trifolii*, (G) diving beetle, (H) *Notonecta Undulate*, (I) damselfly. Scale bars, 100 μm .

1.3.2 Evolution of mushroom bodies

Mushroom body like structures have been discovered in many ecdysozoan lineages including chelicerates (e.g. spiders, scorpions), diplopods (millipedes), chilopods (centipedes), Onychophora and many non-insect hexapods (Strausfeld et al., 1998). Notably, crustaceans do not have mushroom bodies, instead they possess hemiellipsoid

bodies, which are composed of globuli-like cells but lack a proper stalk-like pedunculus (Strausfeld et al., 1998). Whether hemiellipsoid body of crustaceans is a degenerated mushroom body remains a highly debated issue. Moreover, mushroom bodies like structure have also been described in some Lophotrochozoan lineages – Annelids (e.g. *Platynereis dumerilii*) and polyclad Platyhelminths. Do the mushroom bodies found in annelid and insect brains share common ancestry? Or did they evolve independently as a result of convergent evolution?

The size and shape of mushroom bodies varies from species to species, often in correlation with behavioral complexities (Farris, 2005; Farris and Roberts, 2005; Strausfeld et al., 1998). Figure 4 (based on (Strausfeld et al., 1998)) shows a schematic comparison of the structure of mushroom bodies in odor-sensitive and odor-insensitive insect species. Although the number and shapes of calyces, lobes and pedunculi vary from species to species, the basic ground plan (i.e. clustered small-diameter globuli cells and stalk-like pedunculus) is shared (Strausfeld et al., 1998). In addition, the sensory inputs received by the mushroom bodies also show variations. For instance, in honey bees the mushroom body calyces receives visual input from the optic lobes (Jawlowski, 1958; Jawlowski, 1960) in contrast to other insects orders, suggesting an evolutionary adaptation of mushroom body neuronal network in honey bees to specific visual requirements.

Many studies have suggested that the mushroom bodies evolved many times independently. To quote (Farris, 2005), “*Considering the lack of mushroom bodies in the most basal hexapod lineages and assuming that crustacean hemiellipsoid bodies are not homologous to mushroom bodies, it seems likely that mushroom bodies arose independently more than once in the invertebrates.*” However, so far most of the comparisons have been done at structural level only. Further detailed investigations of cell type composition and gene regulatory networks involved in mushroom bodies development in slow-evolving species (e.g. *Platynereis dumerilii*, onychophorans) are required to elucidate the evolutionary origin of mushroom bodies.

1.3.3 Comparison between mushroom bodies and vertebrate brains

Since the discovery of mushroom bodies in 1850, numerous parallels have been drawn between mushroom bodies and vertebrate brain centers. Dujardin (Dujardin, 1850; Strausfeld et al., 1998) himself was intrigued by the similarities of mushroom bodies and the folds and gyri of the cerebral cortex. Later, Hanström (Hanström, 1928) suggested that mushroom bodies are analogous to the vertebrate thalamus. And more recently, the parallels have been drawn between mushroom bodies and vertebrate hippocampus, based on the observations that both mushroom bodies and hippocampus are involved in similar kinds of learning and memory functions (Mizunami et al., 1998b). Additionally, several genes, involved in memory formation mechanisms, are over-expressed both in *Drosophila* mushroom bodies and vertebrate hippocampus (reviewed in (Kandel and Abel, 1995)), further corroborating their analogy. Furthermore, the mushroom bodies have also been compared to the vertebrate cerebellum, the striate cortex and olfactory cortex, mainly based on the cellular arrangements and similar involvement in the olfactory neuronal pathway. Yet another interesting similarity has been proposed at sub-structural organization level of mushroom bodies and vertebrate layered cerebral cortex (Farris, 2005; Farris, 2008). The vertebrate cerebral cortex is a layered structure and segregates the functional connections into six distinct layers, in such a way that the cortical layers II and III specialize in cortico-cortical integration, layer IV receives input connections and layers V and VI generate output connections. Similarly, in mushroom bodies, the input connections are received at the calyces, the pedunculi integrate the information, and the lobes generate output connections. Though all this happens at the level of single Kenyon cells, the basic functional segregation of structured sensory input, information integration and complex behavioral output is analogous to the situation in vertebrate cerebral cortex. As discussed above, most of the comparisons between the mushroom bodies and the vertebrate brain have been based on anatomical characteristics and the functional equivalency of involvement in memory formation. It is still not clear if these observed analogies are indicative of homology (shared common ancestry) or of homoplasy (independent origins), with the latter being the preferred interpretation in the majority of

cases. A better way to address this question further will be to compare at the level of cell types that are present in mushroom bodies and vertebrate brains.

1.3.4 Molecular fingerprint of insect mushroom bodies

Most molecular studies on mushroom body development so far have been carried out in *Drosophila*. The mushroom bodies in *Drosophila* develop from four neuroblasts (MBNBs) which are characteristically located on the vertex of the embryonic and larval brain hemispheres (Ito and Hotta, 1992; Prokop and Technau, 1991; Truman and Bate, 1988). By late embryonic stages, the peduncle and the lobes can already be recognized morphologically, due to the presence of a considerable number of neurons. In the adult mushroom bodies, three types of neurons have been described, namely $\alpha\beta$, $\alpha'\beta'$ and γ . (Crittenden et al., 1998; Ito and Awasaki, 2008) has shown that all four MBNBs contribute to the three neuron types of adult mushroom bodies.

Several studies in *Drosophila* have identified many genes involved in the proper development of mushroom bodies. Firstly, (Kurusu et al., 2000; Noveen et al., 2000; Urbach and Technau, 2003) have reported that mushroom body neuroblasts in *Drosophila* are uniquely defined by the combination of *Dach*, *Pax6*, *BFI* and *Svp*. Additionally, *Eya* and *So* are not expressed in mushroom body neuroblasts (Kurusu et al., 2000; Noveen et al., 2000), and since *Eya* and *So* are part of the eye specification network (alongwith *Dach* and *Pax6*), the absence of expression of *Eya* and *so* makes the mushroom body neuroblasts different from the eye precursors. Also, functional interference experiments with *Dach* and *Pax6* demonstrate that their expression is essential (Kurusu et al., 2000; Noveen et al., 2000) for the proper development of mushroom bodies. On the other hand, over-expression of *ey* in mushroom bodies with the help of a mushroom body specific promoter resulted in the opposite phenotype. Moreover, *dach* expression is not reduced in mushroom bodies cells in *eyless* knockout experiments and conversely, *ey* expression is not decreased in *dach* knockout experiments. This suggests that *dach* and *ey* do not regulate each other during mushroom body development, which is in contrast to their role in the eye specification gene regulatory network. Additionally, a number of other marker

genes have been identified that are expressed in mushroom body cells. In summary, the molecular fingerprint of mushroom body cell types consists of the following genes:

(i) Transcription factors: *Ascl*(Urbach and Technau, 2003), *BF1/Slp*(Urbach and Technau, 2003), *Brf* (Kobayashi et al., 2006), *Dach*(Kurusu et al., 2000; Noveen et al., 2000), *fkh*(Kobayashi et al., 2006), *Hr46*(Kobayashi et al., 2006), *jing*(Kobayashi et al., 2006), *Lhx2*(Herzig et al., 2001), *Otx*(Urbach and Technau, 2003), *Pax6/ey*(Kurusu et al., 2000; Noveen et al., 2000), *Rx*(V. Hartenstein unpublished result in (Hartenstein, 2002)), *Svp*(Urbach and Technau, 2003), *tll*(Urbach and Technau, 2003)

(ii) Neurotransmitters: glutamate, aspartate, taurine(Farris, 2005), neuropeptide F (Johard et al., 2008), FMFRamide (Honey Bee, Cockroach)(Farris, 2005), GCCK(Farris, 2005), Nitric Oxide (NAPDH histochemistry analysis)(Farris, 2005).

(II) Neurotransmitter Receptors: Serotonin Receptors (Yuan et al., 2006), Acetyl Choline Receptors (Cayre et al., 1999), GABA receptors(Cayre et al., 1999), Dopamine receptors (DAMB)(Crittenden et al., 1998), Octopamine receptor (OAMB) (Han et al., 1998)

(iii) Signaling Molecules: Wnt5 (Grillenzoni et al., 2007).

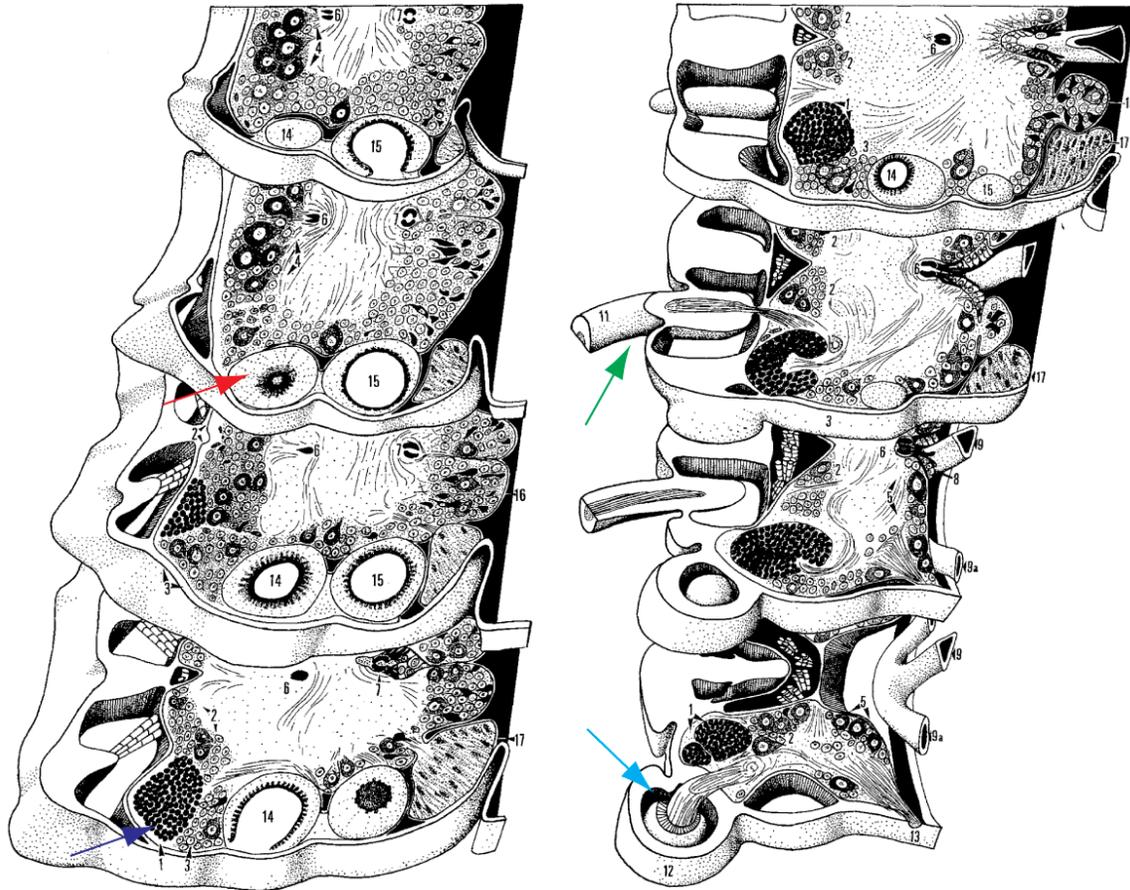


Figure 5. The organization of the adult *Platynereis dumerilii* brain.

The figure is adapted from (Muller, 1973). It shows the schematic drawings of a series of sections of left half of adult *Platynereis* brain in dorsal (Top-left) to ventral (bottom-right) orientation. Eyes: Red arrow, Corpora pedunculata (mushroom body): Blue arrow, Antenna: Green arrow and Palpae: Cyan arrow.

1.4 Mushroom body like structure in *Platynereis dumerilii* brain

Mushroom body like structures have been identified in many annelid species including *Platynereis dumerilii* (Muller, 1973), *Nereis diversicolor* (Heuer and Loesel, 2008; Strausfeld et al., 1998; Strausfeld et al., 2006), Scale worms (Strausfeld et al., 1998) and Sabellida worms (Strausfeld et al., 1998). Classically, these mushroom body-like structures have been termed *corpora pedunculata*. Figure 5 (based on (Muller, 1973)) shows the schematic organization of the head of an adult *Platynereis* worm. The mushroom bodies (marked by blue arrow in Figure 5) in *Platynereis* brain is composed of

tightly clustered globuli-like cells located anterior to the eyes, lateral to antenna and dorsal to palpa. However, it is evident that the schematics reported in (Muller, 1973) is of low resolution, as the details of mushroom body pedunculi and lobe are not resolved. Moreover, there is no information on the input and output connectivity of the *Platynereis* mushroom bodies. Therefore, to compare the anatomy of mushroom bodies in *Platynereis* and insects in more detail, further structural investigations are needed.

1.5 Vertebrate telencephalon patterning

1.5.1 Vertebrate telencephalon development

The vertebrate telencephalon originates from the anterior end of the neural plate (Hebert and Fishell, 2008). Figure 6 summarizes the telencephalon development process in mouse. Initially, the entire neural plate has anterior character and expresses many anterior marker genes. These marker genes are subsequently restricted to the anterior part of the neural plate by the antagonistic interactions of signals originating from the organizer and the anterior visceral endoderm (AVE)(de Souza and Niehrs, 2000; Foley and Stern, 2001; Lu et al., 2001). A number of molecules, including Wnts, FGFs and retinoic acid, act as posteriorizing signals(Altmann and Brivanlou, 2001; Moon and Kimelman, 1998; Sasai and De Robertis, 1997; Schier, 2001) to suppress the anterior character of the neural plate. Many antagonists (including dickkopf and Cerberus) of these posteriorizing signals are expressed in the anterior part to maintain the anterior character of the neural plate. These antagonistic interactions lead to the specification of prosencephalon (forebrain), which is further subdivided into telencephalon and diencephalon by the graded expression of *Wnts* and their antagonist *Tlc*. Once the telencephalon primordium has been specified, it is further subdivided into distinct territories by the actions of morphogens including *Wnts*, *FGFs*, *Bmps* and *Sonic Hedgehog*, which establishes the coordinate information and leads to the expression of region-specific transcription factors. The activities of these transcription factors then give rise to distinct progenitor domains – dorsal cortical ventricular zone and the vertical eminences. Subsequently, these progenitor domains produce specific cell types to yield the mature telencephalon.

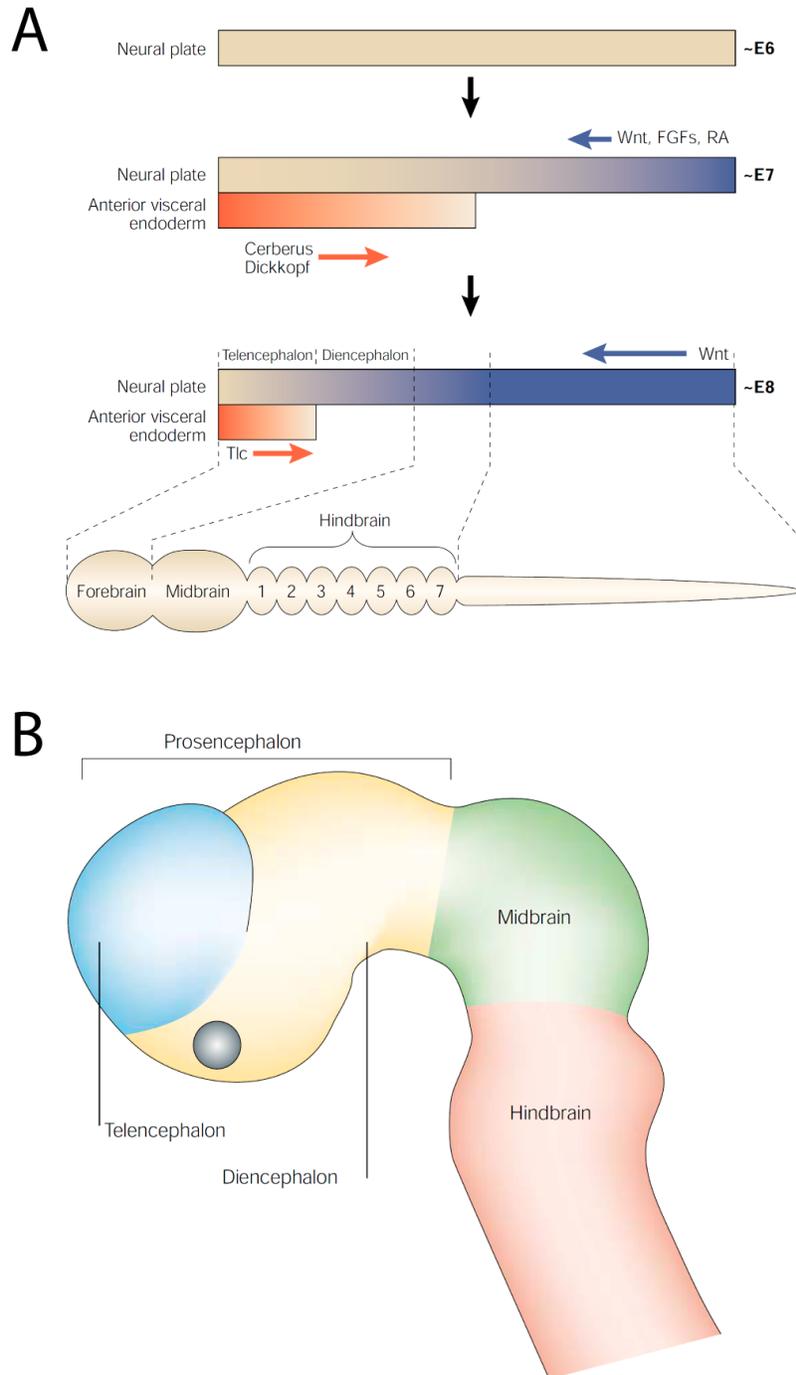


Figure 6. Development of vertebrate brain.

(A) Initially the neural plate expresses several anterior markers which will be further restricted to the anterior domains by the posterior character promoting effects of Wnts, FGFs and retinoids(RA). On the other hand, antagonists of these factors, cerberus and dickkopf, are expressed in the anterior part to preserve the anterior neuronal character. Later in development, the anterior neural plate is further subdivided by the graded expression of Wnts and their antagonists Tlc. E: Embryonic day. (B) Lateral view of the embryonic brain of Mouse at Embryonic day 10 (E10). The main subdivisions are Prosencephalon (including telencephalon and diencephalon), Midbrain and Hindbrain. The figure is adapted from (Rallu et al., 2002)

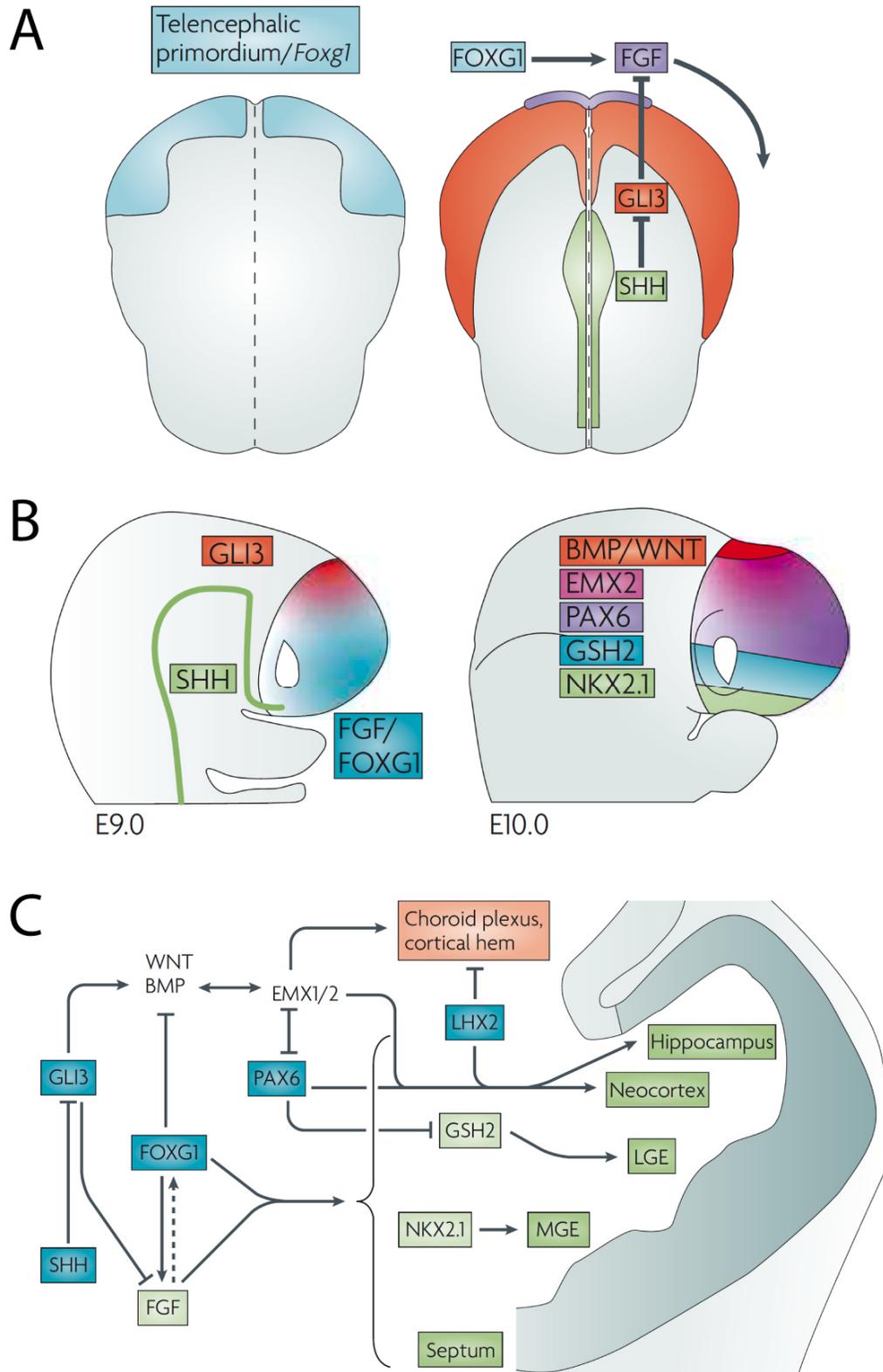


Figure 7. Dorsal-ventral patterning of telencephalon.

(A) dorsal view of the anterior neural plate in a mouse embryo at 5 somites stage. The telencephalic primordium is specified by the expression of FOXG1 (BF1; blue). The nascent telencephalon is patterned

by the interaction between Foxg1, sonic hedgehog (SHH; green) and fibroblast growth factor (FGF; purple). Foxg1 directly promotes the expression of FGFs whereas sonic hedgehog inhibits the repressor activity of GLI3 (dorsalizing factor) to promote the expression of FGFs indirectly. (B) The schematic shows the subdivisions of mouse telencephalon into dorsal and ventral domains at embryonic day (E) 9 and subsequently into four domains by E10. The anterior is right, dorsal is up and ventral is down. At E9.0 Gli3 is expressed dorsally. By E10, the telencephalon is subdivided into four domains by the overlapping gradients of expression of Emx, Pax6, Gsh2 and NKX2.1. The expression domain of Gsh2 overlaps with that of Nkx2.1 and is not shown in the schema for simplicity. Also, the expression of Foxg1, Shh and FGFs is not shown in the schema of E10.0 for the sake of clarity (C) The schematic representation of the gene regulatory network involved in early specification of telencephalon. LGE: Lateral Ganglionic Eminence, MGE: Medial Ganglionic Eminence. All the three panels (A,B,C) are adapted from (Hebert and Fishell, 2008).

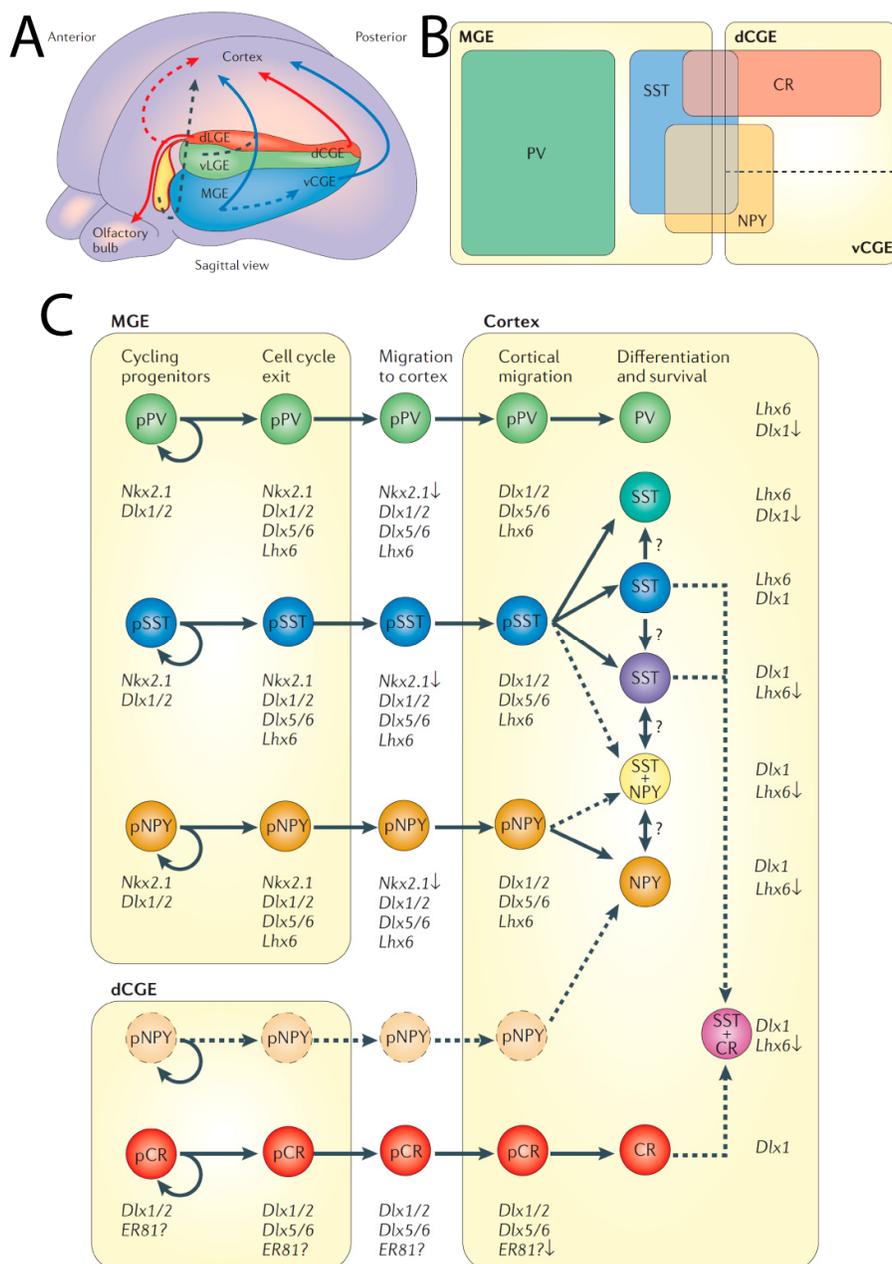


Figure 8. The cell types emerging from ventral telencephalon.

The ventral telencephalon consists of LGE (Lateral Ganglionic Eminenc), MGE (Medial Ganglionic Eminence) and CGE (Caudal Ganglionic Eminence). These regions give rise to several subtypes of GABAergic interneurons. (A) Schematic representation of migrating interneurons from ventral telencephalon to cerebral cortex. Blue arrows marks the migration pathways of somatostatin (SST) or parvalbumin (PV) containing interneuron progenitors. Red arrows mark the migration pathways of calretinin (CR) containing interneuron progenitors and black arrows marks the migration of other, yet unknown, subtypes of interneurons. (B) Schematic representation of the relative contributions of MGE and CGE to the cortical interneuron cell types diversity. Parvalbumin containing interneurons mainly originates from MGE, somatostatin containing interneurons mainly originates from MGE, neuropeptide Y containing interneurons originates from both MGE and CGE, whereas calretinin containing interneurons mainly originates from dorsal CGE. (C) Schemataic representation of gene networks involved in regulating the fate specification of cortical interneurons. (↓) represents the downregulation. The figure is based on (Wonders and Anderson, 2006).

1.5.2 Telencephalon patterning and the emerging cell types

The earliest marker of the telencephalon anlagen is a forkhead box transcription factor *foxf1* (*bfl*, Brain Factor1) (Tao and Lai, 1992) (Figure 7A). Its expression starts when the telencephalon primordium is still a one cell thick neuroepithilium. Shortly after the onset of *bfl* expression, the telencephalon is divided into dorsal and ventral domains by the dorsalizing effect of *gli3* expression (Grove et al., 1998; Theil et al., 1999; Tole et al., 2000) and ventralizing effect of *shh* expression (Chiang et al., 1996). Subsequently, the dorsal telencephalon is subdivided into a medial region expressing *BMPs* and *Wnts* and a lateral region expressing *Emx* and *Pax6* in antagonizing gradients. The ventral telencephalon is subdivided into lateral *Gsh* expressing domain and medial *nk2.1* expression domain. Subsequently, *nk2.1* expression extends into the *Gsh* expression domain.

Later in development, the dorsal telencephalon (also called pallium) gives rise to mainly glutamatergic neurons and structurally subdivides into antero-laterally positioned neocortex and postero-medially located hippocampus, cortical hem and choroid plexus (Hebert and Fishell, 2008). The ventral telencephalon also subdivides into two main structures – a medial region, the medial ganglionic eminence (MGE), and lateral regions, the lateral ganglionic eminence (LGE) and caudal ganglionic eminence (CGE). The ventral telencephalon gives rise to many subclasses of GABAergic neurons Figure 8 (Wonders and Anderson, 2006). The medial ganglionic eminence (MGE) gives rise to *somatostatin*, *parvalbumin* and *npy* (Neuropeptide Y) expressing subclasses of

GABAergic neurons, the caudal ganglionic eminence (CGE) produces *calretinin* and *vip* (Vasoactive intestinal peptide) expressing interneurons and the lateral ganglionic eminence (LGE) produces inhibitory and most of the olfactory bulb interneurons. Many of these distinct cell types migrate to make complex cerebral cortex, basal ganglia and limbic structures including amygdala and the nucleus accumbens (Figure 8A).

Many studies over the years, utilizing loss of function and gain of function experiments in the vertebrate model systems, have identified the gene regulatory interactions involved in patterning the telencephalon. Figure 7C summarizes the gene regulatory interactions involved in telencephalon patterning. For the specification of dorsal telencephalon, *gli3* activates the expression of *Wnts* and *Bmps*, which in turn activate *Emx* expression. Together with *Pax6* and *Lhx2*, *Emx* further subdivides the telencephalon into neocortex and hippocampus. For the ventral telencephalon specification *shh* represses the dorsalizing effect of *gli3* and together with *bfl/foxg1* activates the expression of *fgfs*. Further, *bfl/foxg1* together with *fgfs* activates the downstream transcription factors including *gsh* and *nk2.1* to specify the LGE and MGE.

1.5.3 Evolution of telencephalon

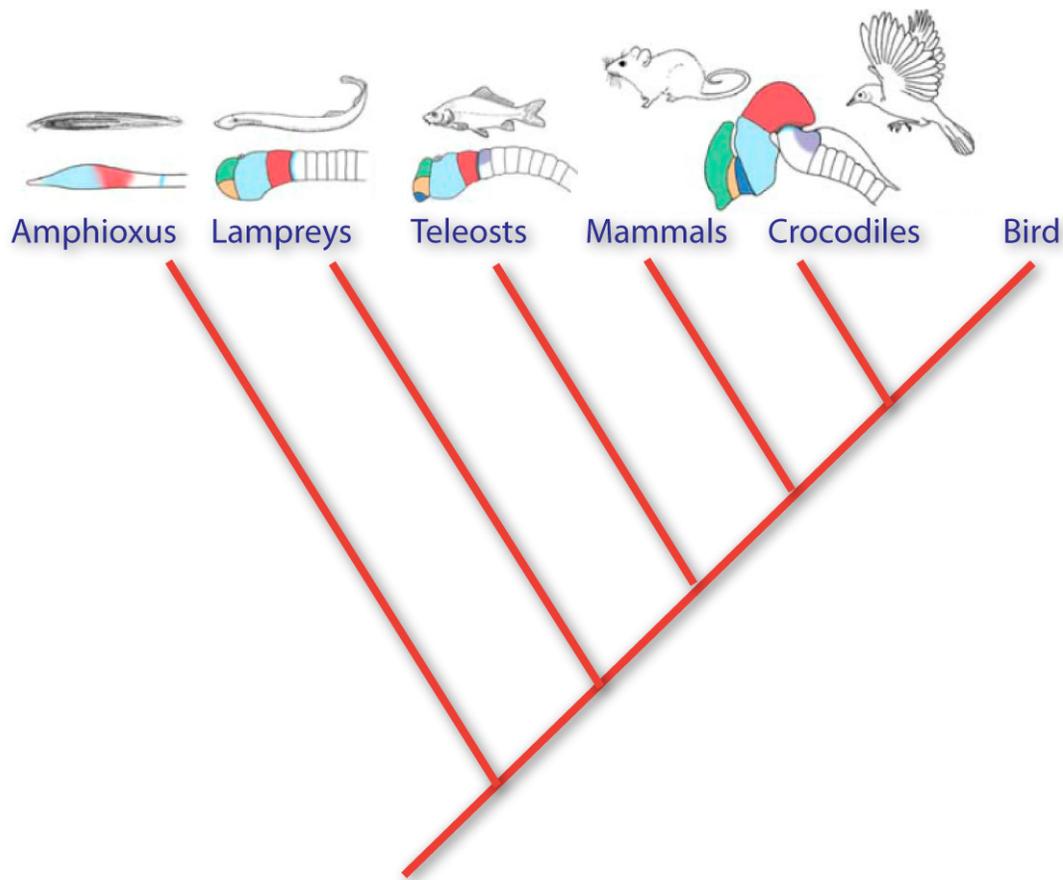


Figure 9. Evolution of telencephalon.

The schematic shows the evolutionary scenario of vertebrate telencephalon. Pallium (dorsal telencephalon) is shown in green, Lateral ganglionic eminence is shown in orange and Medial ganglionic eminence (MGE) is shown in blue. The basal chordate amphioxus does not possess a morphologically recognizable telencephalon. The figure is adapted from (Murakami et al., 2005).

The relative size and complexity of telencephalon vary in each lineage in vertebrates. Figure 9 summarizes the comparison of the telencephalon in vertebrates and a basal chordate amphioxus (cephalochordate) (Murakami et al., 2005). The three major domains of the telencephalon are marked - pallium (dorsal telencephalon) in green, LGE (Lateral ganglionic eminence) in orange and MGE (medial ganglionic eminence) in blue. Using morphological criteria and molecular marker expression analysis, the existence of a telencephalon is shown in all the vertebrate lineages. Most of the vertebrates possess all the three major domains (pallium, LGE and MGE) of the telencephalon. However, lampreys (jawless fish) lack a visible MGE suggesting either the MGE was secondarily

lost in lampreys or the MGE evolved after the split with lampreys, the later being the favorite. On the other hand, existence of telencephalon in amphioxus (a basal chordate) is still disputed. The anterior end of the amphioxus neural tube contains slightly dilated structures called cerebral vesicle (shown in Cyan in Figure 9). Are at least parts of the amphioxus cerebral vesicle homologous to the vertebrate telencephalon or diencephalon? Morphological comparative studies have not found many similarities. The current consensus is that the amphioxus cerebral vesicle corresponds to the vertebrate diencephalon (Murakami et al., 2005). However, interestingly, amphioxus cerebral vesicle expresses many telencephalic markers including *BFI*, *Dlx*, *Pax6* and *Otx*, raising the possibility of mixed population of cell types in the cerebral vesicle (Benito-Gutierrez, 2006).

1.6 Aim of the thesis

The aim of my PhD thesis was threefold:

1. To establish a computational protocol, utilizing image processing algorithms, for generating high-throughput gene co-expression information. This protocol would allow me to describe the molecular fingerprint of various cell types present in the *Platynereis dumerilii* larval brain.
2. To investigate the evolution of mushroom bodies in Protostomes. For this, I aimed to describe the structure and development of mushroom bodies in *Platynereis dumerilii* and subsequently identify the molecular fingerprint and compare with the insect mushroom bodies.
3. To compare the mushroom body cell types with functionally equivalent cell types in vertebrates' forebrain.

2. RESULTS

2.1 Whole Mount In Silico Expression Profiling

2.1.1 The WMISEP Protocol

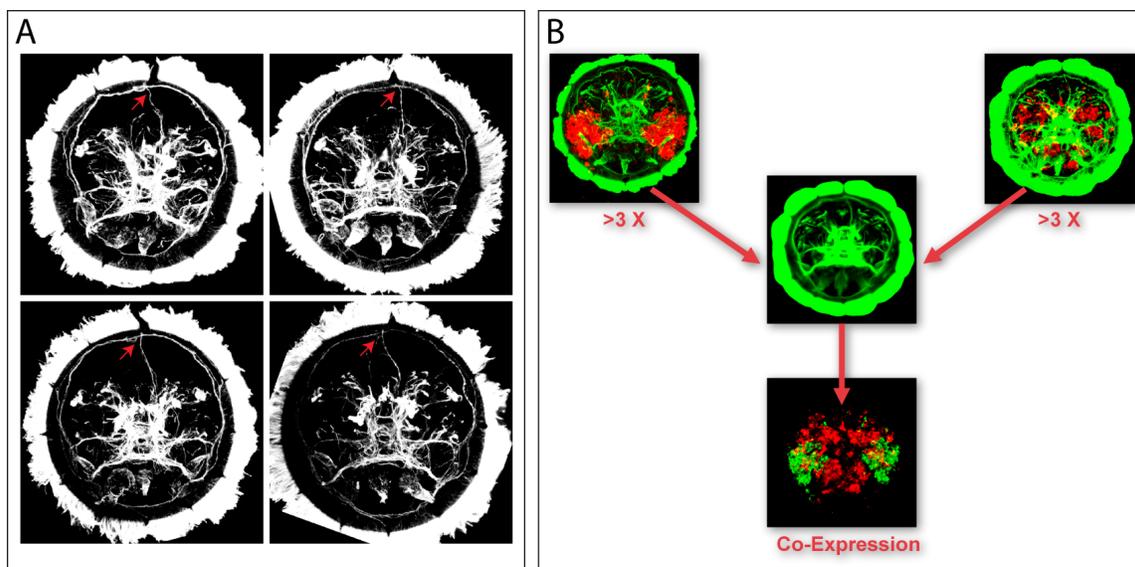


Figure 10. The basic idea of Whole Mount In Silico expression Profiling (WMISEP) protocol.

(A) Comparison of axonal scaffold of four different 48 hour old *Platynereis* larval brain. Confocal microscopy was used to acquire 3D image stacks of larvae stained with an antibody against acetylated tubulin to visualize the axonal scaffold. Maximum Z-projections of the image stack are shown. All the pictures are from apical view, dorsal is up and ventral is down. Red arrows mark a single asymmetric neuron (which connects the medial brain to dorsal prototroch ring), as an example of stereotypic nature *Platynereis* embryonic development. (B) The basic principle of the WMISEP protocol is to acquire two-color confocal image stacks, with red channel containing spatial expression information for a gene, acquired using reflection confocal microscopy of NBT/BCIP precipitate and the green channel containing information of the axonal scaffold visualized using immunostaining against acetylated tubulin. The information in the green channel is then used to align the 3D images to a common reference average axonal scaffold image, and thus bringing the expression patterns into the same coordinate system.

2.1.1.1 The Basic idea

Platynereis embryonic development is stereotypic and synchronous, implying that embryos of similar age are similar in terms of anatomy and developmental stage. Figure 10A shows the comparison of the axonal scaffolds of the brains of four different 48 hours old *Platynereis* larvae, visualized using an antibody against acetylated tubulin. It is evident that the four images are very similar, confirming the stereotypic nature of

Platynereis embryonic development. Looking in further details in 3D suggests the similarities even at the single axon level. For instance, an asymmetric neuron (marked by red arrow in Figure 10A), which connects the medial brain to dorsal prototroch ring, is found to be present in similar spatial locations in all the four different larvae examined in Figure 10A. This observation raises some obvious questions: Can the 3-dimensional images of two or more *Platynereis* larval brain be aligned computationally? And can it be used to generate high-throughput gene co-expression information?

Utilizing the stereotypic features of *Platynereis* embryonic development, I developed a protocol (Whole Mount In Silico Expression Profiling, WMISEP) to generate high-throughput gene co-expression information. The basic idea behind the protocol is summarized in Figure 10B. Two 3D images of 48hpf larvae are acquired, containing information in two colors - Green and Red. The green channel contains the information about the axonal scaffold of *Platynereis* larval brain, visualized using an antibody against acetylated tubulin. The red channel contains the gene expression information acquired using Whole Mount Reflection Microscopy (Jekely and Arendt, 2007). The information in the green channel is utilized to align the two two-color 3D images to a third 3D reference image resulting in a composite image which is then used to generate spatial gene co-expression information. The advantage of aligning images to a common reference image rather than to each other is that all the aligned images are brought into the same coordinate system, allowing for any comparison among them.

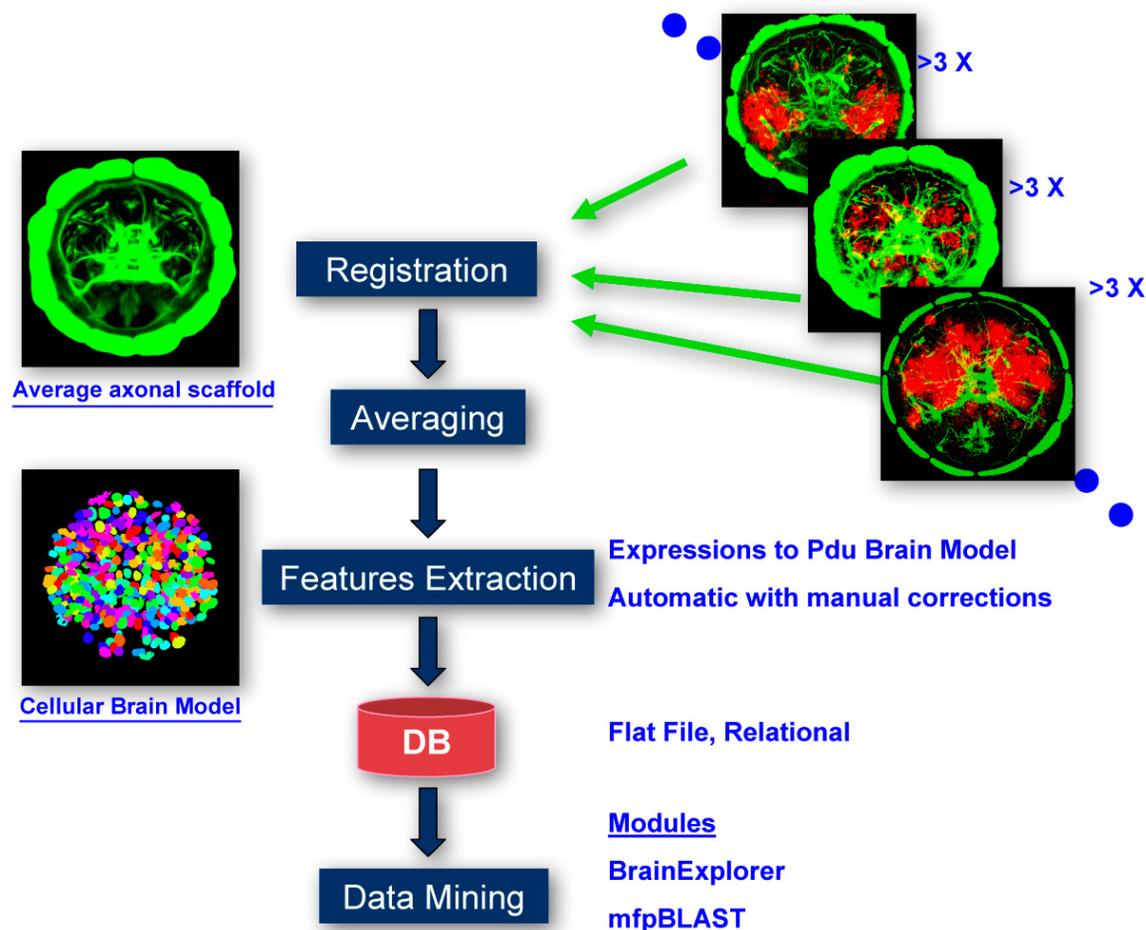


Figure 11. Overall summary of Whole Mount In Silico Expression Profiling (WMISEP) Protocol.

The confocal microscopy is used to acquire two-color image stacks in such a way that the red channel contains the spatial expression information of a gene and the green channel contains information of the axonal scaffold. The information in the green channel is used to align several such images to a common reference average axonal scaffold image, and thus bringing the expression patterns of several genes into the same coordinate system. The expression patterns are averaged by combining the information from multiple embryos stained with the same gene, which are then superimposed on a cellular brain model to yield cellular gene expression profiles. This information is fed into a database, which could be searched in a flexible manner. For instance, mfpBLAST can be used to identify mushroom body anlagen cells by searching for cells that express *Dach*, *Pax6*, *Svp*, and *BFI* but do not express *Eya* and *So*.

2.1.1.2 Overview of WMISEP

I developed a computational protocol (Figure 11) to generate high-throughput co-expression information for many genes. The first step in the protocol is to acquire two-channel 3D images using confocal microscopy. One channel contains the information about the axonal scaffold of *Platynereis* larval brain and the other channel contains the spatial expression information for a gene. For every gene, multiple spatial expression

images are acquired from different stained larvae. These three-dimensional images are then aligned to a reference average brain image, using an image processing algorithm called Image Registration. In the next step, multiple images representing the expression pattern of the same gene are averaged to generate the average spatial expression information. This spatial expression information is then overlaid onto a cellular model of *Platynereis* larval brain, resulting in gene expression profiles at single cell resolution. The expression profile information is then fed into a searchable database. Furthermore, I developed several modules for querying the data, in particular two user friendly interfaces were designed: BrainExplorer and mfpBlast (molecular fingerprint BLAST). In the following sections I have presented the details of all the procedures and several proofs of principle studies of the protocol.

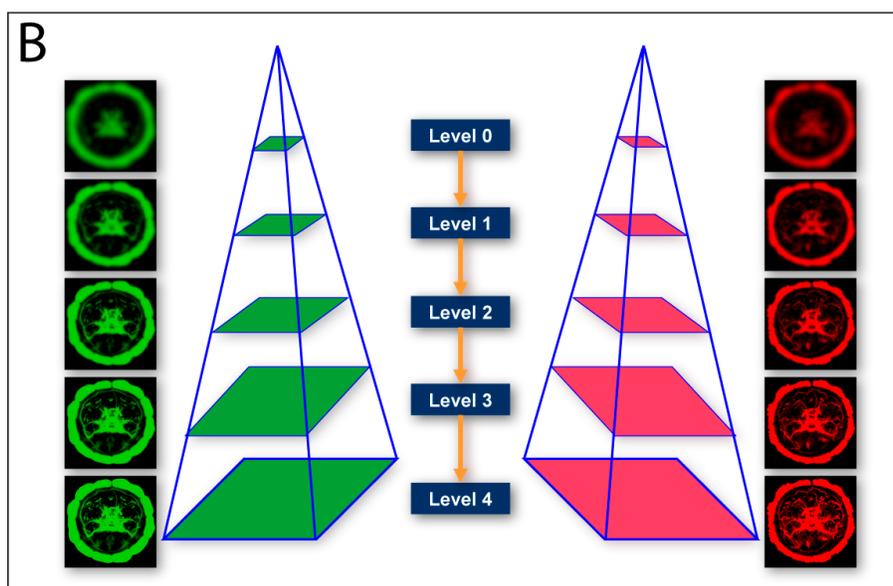
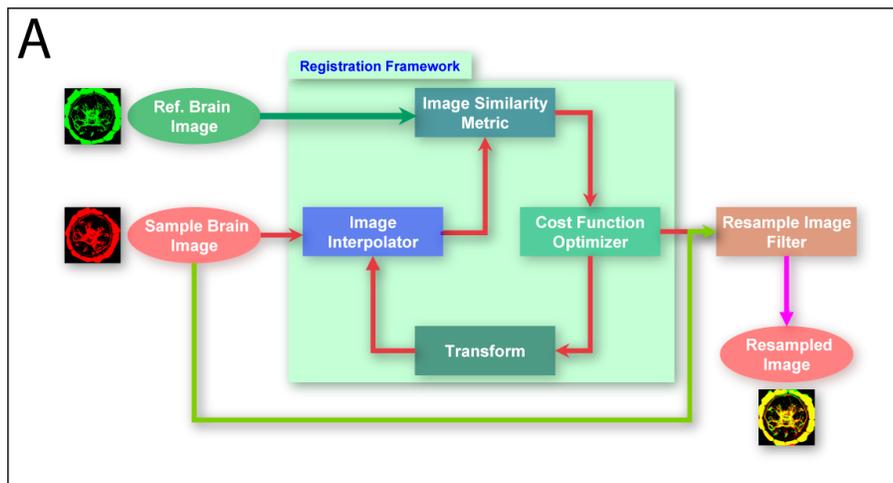


Figure 12. The general workflow of image registration algorithms.

(A) The basic requirements of an image registration algorithm. A metric is needed to assess the image similarities after the alignment; an interpolator is required for estimating the missing intensity values in the confocal image stack; an optimizer is needed to find the optimal value for the cost function; a transformation procedure is needed to rotate, scale, shear or deform the images; and finally a filter is required to resample the images. The arrows mark the direction of information flow. (B) The quality of alignment of two images is significantly improved by aligning them sequentially at several levels of morphological resolution. For WMISEP, the images are consecutively blurred to generate five levels of morphological details. The procedure starts by aligning sequentially from Level 0 (lowest morphological details) to Level 4 (highest morphological resolution) (Ibanez, 2005; Jefferis et al., 2007; Rohlfing and Maurer, 2003).

*2.1.1.3 Registration of *Platynereis* larval brain images*

Image Registration is an image processing procedure to align two multi-dimensional images such that they match maximally. It is a classical problem in image processing field with no general solution which fits for all the applications (Lisa Gottesfeld, 1992). Most of the algorithmic development in this field is motivated by the research done in the Biomedical imaging field. There are many open source projects addressing various issues of the Image Registration problem. Two of the most commonly used and supported frameworks for image registration are National Library of Medicine Insight Segmentation and Registration Toolkit (ITK, www.itk.org) (Yoo et al., 2002) and Statistical Parametric Mapping (SPM, <http://www.fil.ion.ucl.ac.uk/spm/>). ITK is an open source project funded by the National Library of Medicine, NIH to support the Visible Human Project. It provides a C++ library of various fundamental algorithms in Image processing. SPM is an open source project run by the Wellcome Trust Centre for Neuroimaging and is implemented as library of functions in MATLAB (<http://www.mathworks.com/>). SPM is mainly used for analyzing functional imaging (fMRI, PET, SPECT, EEG and MEG) data. Also, there are several other implementations addressing particular problems in the image registration field (For example, AIR (Woods et al., 1992; Woods et al., 1998; Woods et al., 1993), Elastix (Klein et al., 2007; Staring et al., 2007) and the implementations reported by (Jefferis et al., 2007; Rohlfing and Maurer, 2003; Rohlfing et al., 2005)). I wrote several programs (source code given in Appendix, see *wmisp-affine.cxx* for example) to test various algorithms for registering *Platynereis* nervous system images. Finally, I optimized and adapted the multi-processor parallel algorithms and

implementations reported in (Jefferis et al., 2007; Rohlfing and Maurer, 2003; Rohlfing et al., 2005).

In general, an image registration procedure can be summarized as shown in the Figure 12A. There are four basic requirements for an image registration procedure. Firstly, a metric is needed to assess the image similarities after the alignment. There are many measures reported in the literature (reviewed in (Holden et al., 2000)) including: Mean Square difference of intensities, Entropy of the difference image, Mutual information, Pearson product moment cross-correlation and Normalized mutual information. Each of these metrics has advantages and disadvantages. I tested many of them and found that the Mutual Information based metrics give best results for registering *Platynereis* nervous system images. Second, an appropriate Image interpolator is required for interpolating the missing intensity values in the images. As the confocal image stacks is composed of discrete slices, it is essential to estimate the missing values between the slices. There are several interpolators reported in the literature (Ibanez, 2005) including: nearest neighbor, trilinear, polynomial, sinc, B-splines and fourier methods. B-splines and sinc are known to estimate the most smooth intensity interpolations but usually are computationally expensive. I found that the third order B-splines interpolation gives the best result for registering *Platynereis* nervous system images. Third, an appropriate optimizer is needed. Some of the most commonly used optimizers are: conjugate gradient, gradient descent and one plus one evolutionary. The gradients based optimizers work better for aligning *Platynereis* nervous system images. Fourth, an appropriate transformation procedure is needed. I found that the rigid body transformations followed by non-rigid transformations produces the best results for aligning *Platynereis* nervous system images. Also, I tested a multi-resolution approach (Figure 10B) in which 3D images are blurred to various degrees before alignment. The procedure starts by aligning the most blurred image and then moves on to the finer resolution images. I tested various levels of blurring and found that the 5 levels of blurring work optimally for aligning *Platynereis* nervous system images.

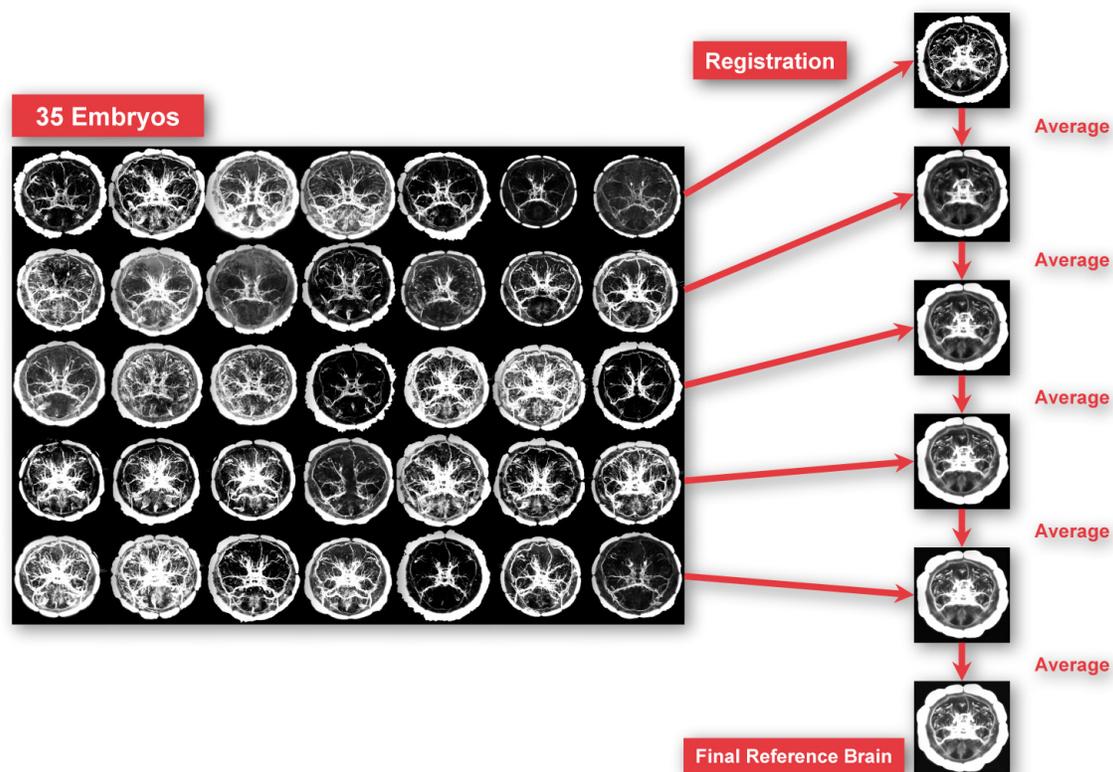


Figure 13. Procedure used to generate the average reference image of axonal scaffold of 48 hour old *Platynereis* larval brain.

The average reference image of the axonal scaffold of *Platynereis* larval brain was generated from 36 individual larval brain images. To begin with, a good looking, in terms of signal strength and spatial orientation, image was used as the starting reference image to which the remaining 35 images were aligned. In the next step, all the resulting aligned images were averaged to generate the first version of an average reference image. Subsequently, all the 36 images from the dataset were aligned to this average image and were again averaged. After iterating this procedure for 5 times, the average image did not change any further. This version 5 average image was used as the reference for all subsequent alignments. All the pictures in the figure are maximum Z-projections of confocal image stacks of axonal scaffold, visualized by immunostaining against acetylated tubulin, of 48 hour old *Platynereis* larval brain.

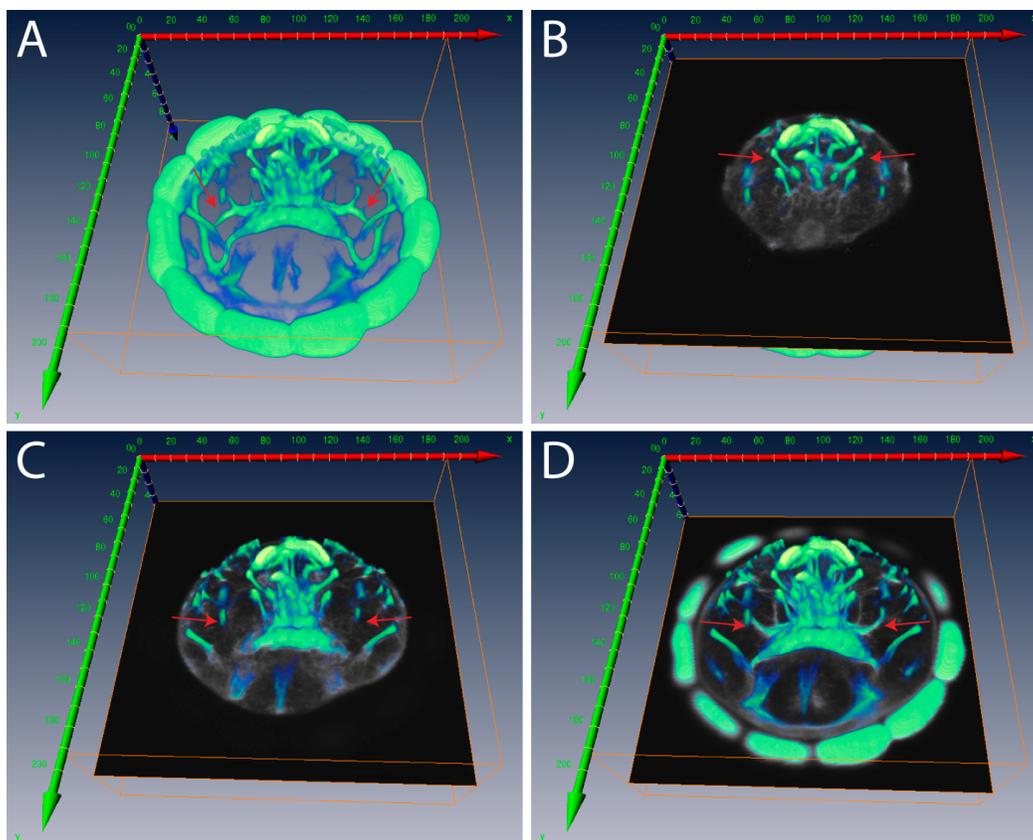


Figure 14. Three dimensional reconstruction of the average reference axonal scaffold of 48 hour old *Platynereis* larval brain.

(A) The three dimensional reconstruction of the average reference larval axonal scaffold was generated using Amira (<http://www.amiravis.com/>). (B-D) illustrates the fine details of the average reference image using a black slider at various optical depths. It is clear that the average reference larval brain image is highly symmetric and has retained majority of the neuronal features. For instance, red arrows in A, B, C and D points towards the symmetrically located connectives, the apical organ tufts, sensory synapses and an unknown connective respectively.

2.1.1.4 Average reference brain image of 48 hour old Platynereis larvae

A good reference *Platynereis* larval brain image is an essential requirement for getting the optimal registration results. I generated an average *Platynereis* larval brain image using images from 36 different individual 48 hpf larvae stained with an antibody against acetylated-tubulin. As summarized in Figure 13, in the first step, I selected the best quality image from the dataset and aligned the rest (35) of the images to it. The resulting aligned images were then averaged to generate a starting average image. Further, I aligned all the starting 36 images to the average image generated in the previous step. The resulting aligned images were again averaged and fed into the next iteration. The

procedure was reiterated until there was no change in the average image generated in the consecutive steps. After five rounds of iterations the average image did not change further. To analyze the resulting image in greater details I did the three-dimensional reconstructions as illustrated in Figure 14. It is evident that most of the features of the *Platynereis* larval brain were recovered from the averaging procedure (examples given in the legend of Figure 14), further confirming the stereotypic nature of the *Platynereis* embryonic development. For all the further alignments, this average three-dimensional image of 48 hour old *Platynereis* larval brain was used as reference.

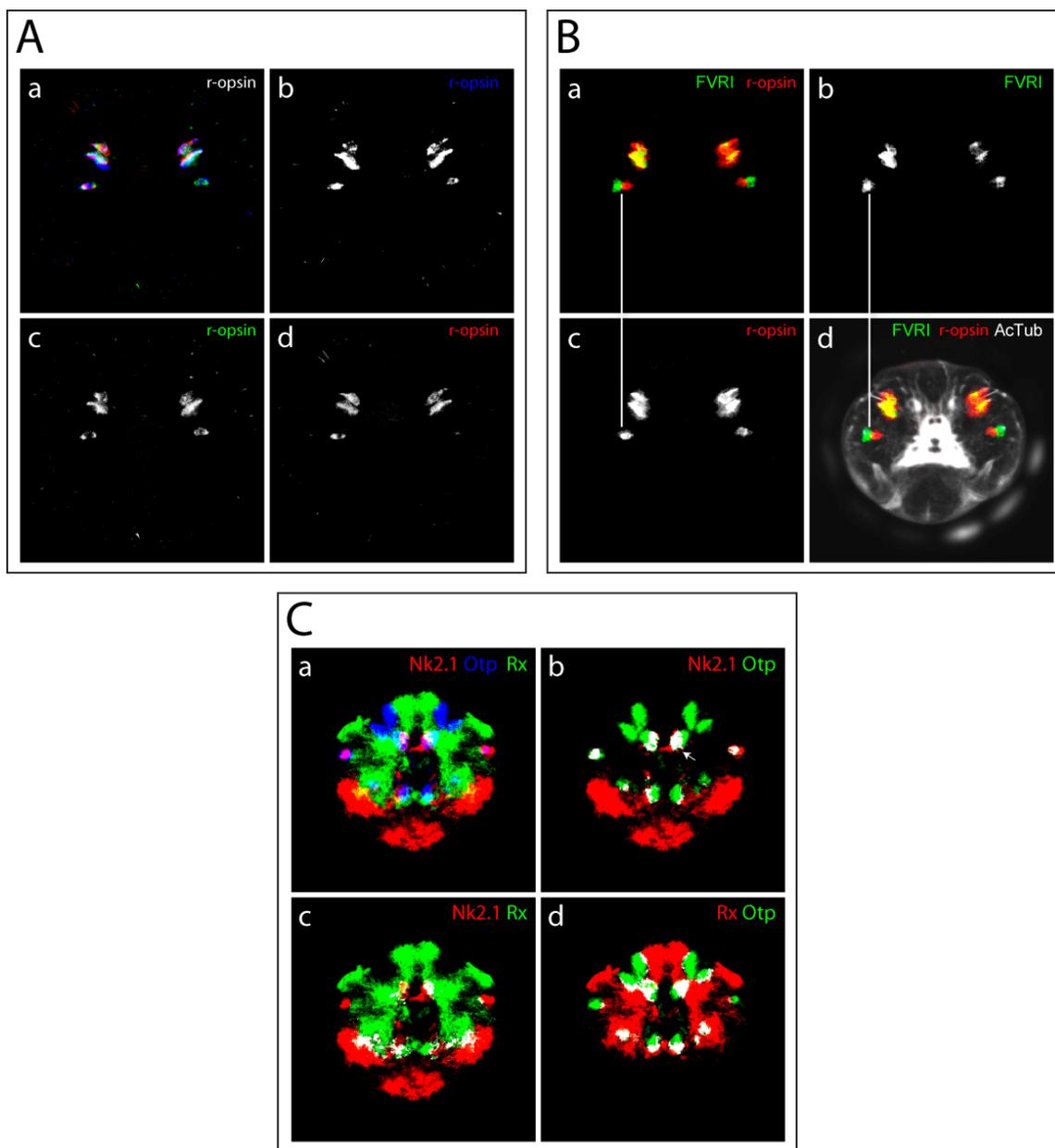


Figure 15. Proof of principle experiments for WMISEP.

(A) A positive control experiment for WMISEP. Confocal images were acquired from three different individual larvae stained for the same gene, *r-opsin*. The images were then aligned using WMISEP. Ab-Ad are the maximum z-projections of three individual images and Aa is the maximum z-projection of the three-color (RGB) merge of these three images. It is clear that the expression signals overlap almost perfectly, in accordance with the expectation. (B) Double combination of *FVRI* and *r-opsin* was generated using WMISEP. The vertical white lines mark the expression of these two genes in two neighboring cells, which is in agreement with the observations made by Keren Guy and Harald Hausen using TEM (Transmission Electron Microscopy) and fluorescence whole mount in situ hybridization. (PhD thesis, Keren Guy). Ba: *FVRI* (green) and *r-opsin* (red). Bb: *FVRI*. Bc: *r-opsin*. Bd: *FVRI* (green), *r-opsin* (red) and immunostaining against acetylated tubulin (white). (C) Triple combination of *NK2.1*, *Otp* and *Rx* was generated using WMISEP. As reported in (Tessmar-Raible et al., 2007), these three genes are co-expressed in a few cells in the medial larval brain. In (perfect) agreement with this observation, the white arrow (in Cb) marks the co-expression. Ca: Maximum z-projection of triple combination of *NK2.1* (red), *Otp* (blue) and *Rx* (green). Cb-Cd Maximum z-projection of the double combinations of *NK2.1* (red) and *Otp* (green); *NK2.1* (red) and *Rx* (green); *Rx* (red) and *Otp* (green) respectively. White marks the co-localized pixels.

2.1.1.5 Proof of Principle experiments for WMISEP

Having established the protocol for WMISEP, the next step was to test the sensitivity and specificity of the protocol. To assess the sensitivity I conducted a positive control experiment in which I acquired images of three different *Platynereis* 48 hour old larvae, which were stained for the same gene (*r-opsin*), using whole mount reflection microscopy. Then I aligned these three-dimensional images to the average reference larval brain image and generated triple combinations in three color channels. Since all the three images contain the spatial expression information for the same gene, the spatial expression signals should overlap with each other, if the technique works accurately. As shown in the Figure 15A, the expression signals indeed overlap almost perfectly at single cell resolution indicative of the high sensitivity of the protocol. Furthermore, I tested for the specificity of the protocol by acquiring the images for the expression of two genes (*r-opsin* and *FVRI*) known to be expressed in two neighboring cells (Keren Guy's PhD Thesis). Subsequently, I generated the combined spatial expression pattern of these two genes using WMISEP protocol. As shown in the Figure 15B, the double combination generated using WMISEP shows that *r-opsin* and *FVRI* are indeed expressed in two distinct neighboring cells. Furthermore, I tested the triple combination of *NK2.1*, *Rx* and *Otp* using WMISEP. These three genes are reported to be co-expressed in a subset of neurosecretory cells in the median brain of *Platynereis* 48 hpf larvae (Tessmar-Raible et al., 2007). As demonstrated in the Figure 15C, I indeed find few cells which co-express

NK2.1, *Rx* and *Otp* and are spatially located in the expected location. Together with these experiments and several other observations (see Figure 42), it is clear that WMISEP is indeed single cell sensitive and high specificity protocol, at least for the brain regions and sensory organs tested.

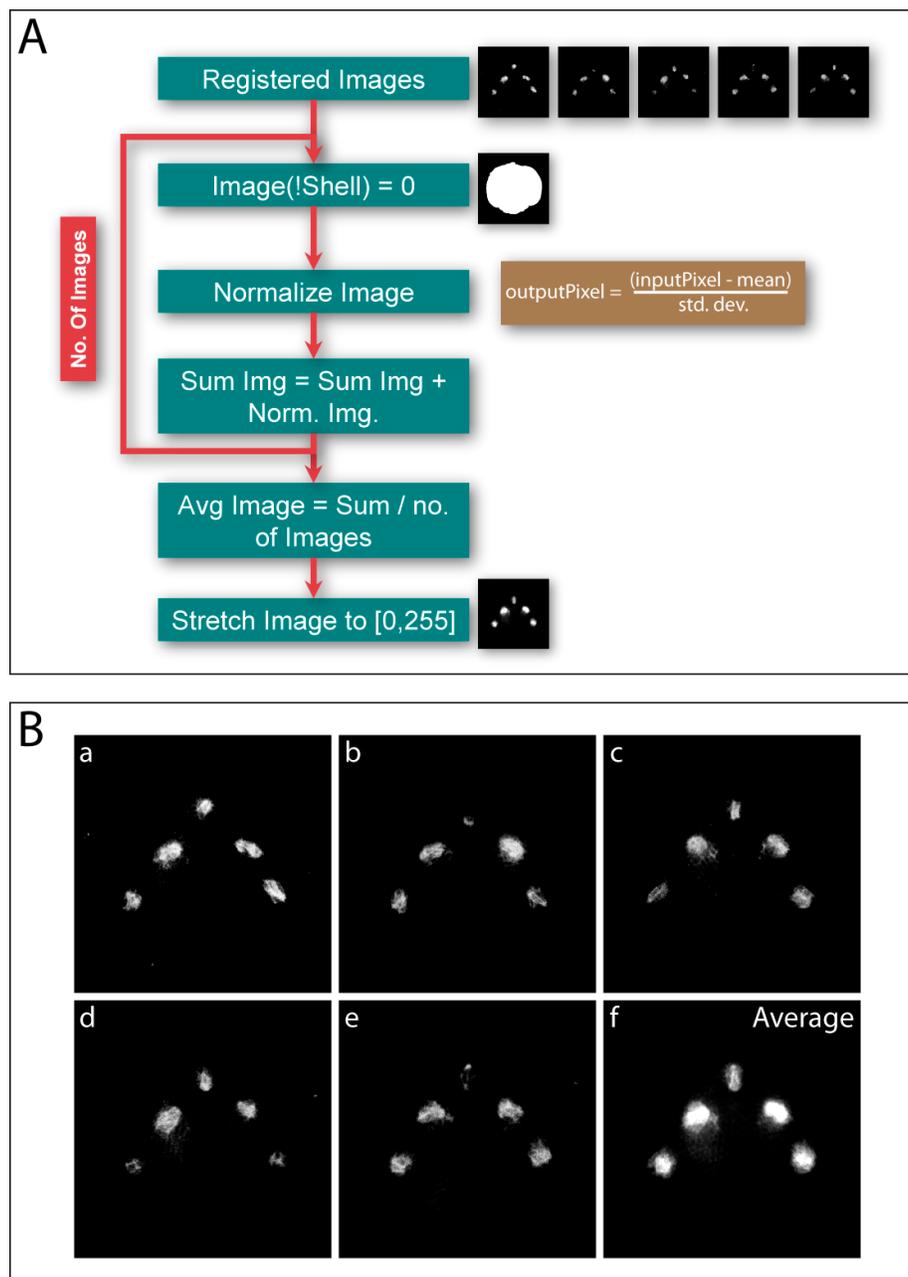


Figure 16. Averaging the images of spatial expression pattern acquired from multiple embryos.

(A) summarizes the algorithm for averaging the expression pattern images. Briefly, the aligned images, generated after registration, are filtered using a shell image in a way that the signal outside the physical

embryo is set to zero. Further, each of the images is normalized in such a way that the mean intensity of the images become 1 and the standard deviation becomes 0. The normalized images are then averaged and transformed back to 8-bit image format. (B) shows an example of averaging the expression patterns. Ba-Be are five different images acquired from five different larvae. Bf shows the normalized average of these images generated using the protocol describes in A.

2.1.1.6 Averaging expression patterns

For directly comparing the spatial expression patterns of several different genes, it is better to calculate the average expression patterns, not to be misled by any peculiarity of an individual larva. Any successful algorithm for generating the average images must take into account these inherent variations present in the individual images. There are a number of sources of variations among the images such as the background noise generated during the experimental staining procedures, gain and offset settings used to acquire the images during the confocal microscopy and the intrinsic biological variability. I designed and tested an algorithm which eliminates some of the inherent variability present in the individual images by normalizing them before averaging. As shown in the Figure 16A, firstly, I developed a 3 dimensional mask (shell) image in such a way that it minimally enfolds the reference brain image. This three dimensional shell is used to eliminate any background signal, outside the larval image, by setting the intensities values to zero. Further, all the images are normalized in such a way that the mean intensity value of the image becomes zero and the standard deviation one to ensure that all the images contribute equally to the final average image. Then the normalized images are added together and averaged by dividing the number of images. The intensity values of the resulting average image are then linearly mapped to the range [0,255] to generate 8-bit images. To illustrate the effect of averaging on the expression pattern, I investigated a gene (*ZFAT*) which has a highly specific expression pattern. I generated five different three dimensional images of the expression pattern of *ZFAT* in five different embryos. As illustrated in the Figure 16B, the averaging procedure indeed takes contribution from all the five images, irrespective of the intensity variability of the individual images. Moreover, the average images are biologically better suited for comparing expression patterns of several genes acquired from different larvae.

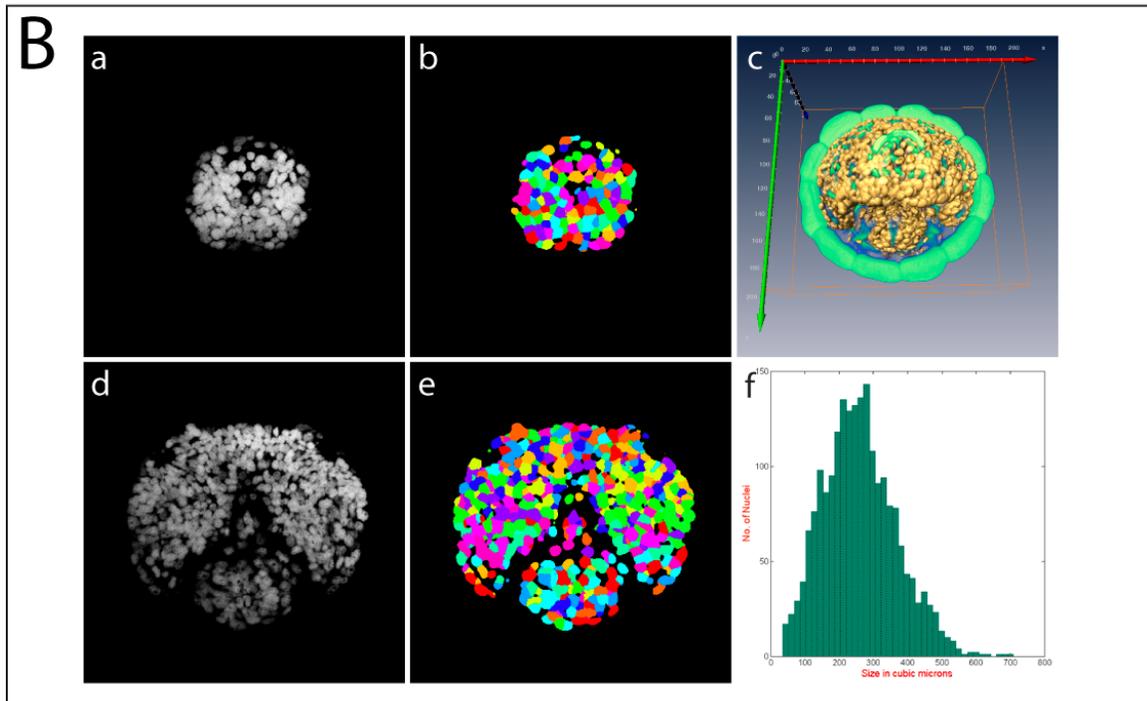
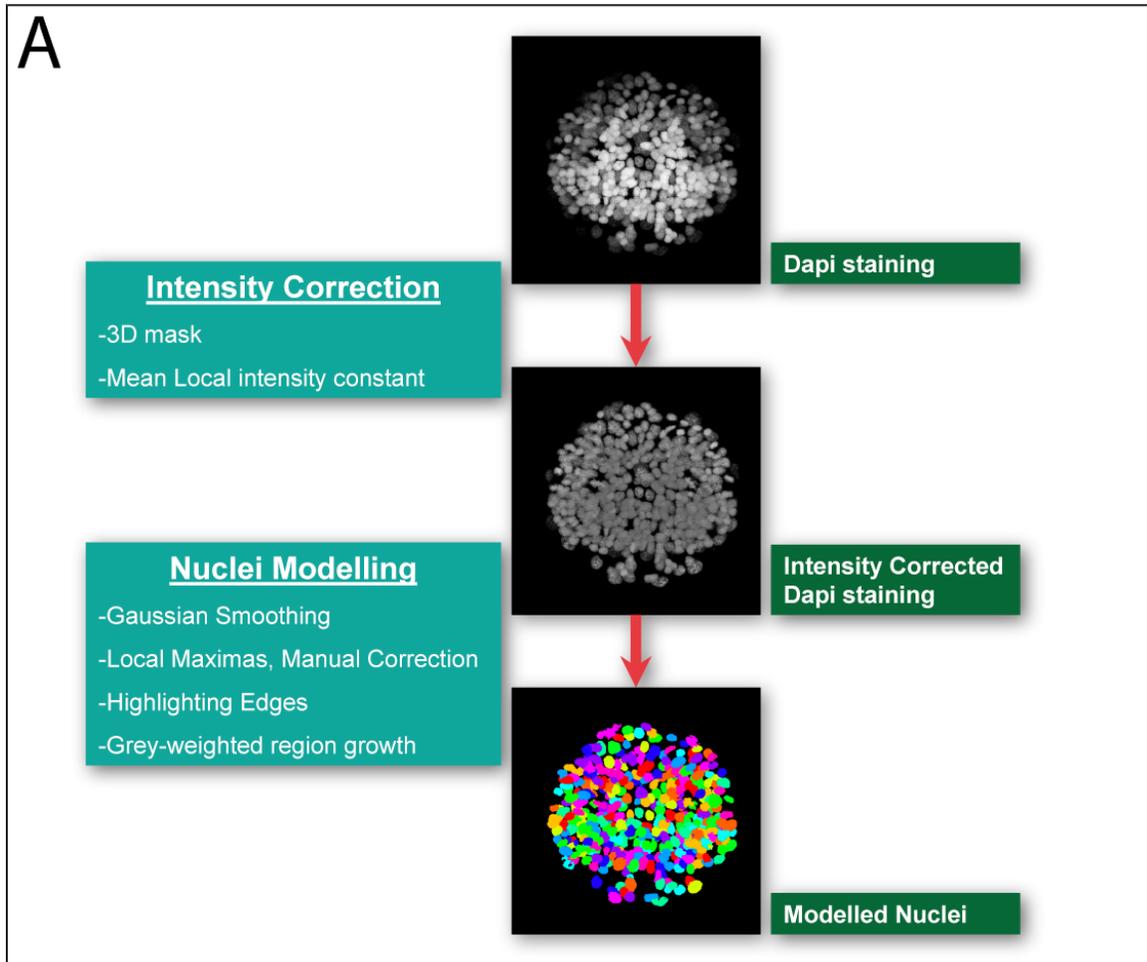


Figure 17. Generating cellular brain model of 48 hour old *Platynereis* larvae.

(A) describes the algorithm used to create the cellular model of *Platynereis* larval brain. Briefly, confocal image stacks were acquired of larvae stained with the nuclear stain dapi to label the nuclei and immunostaining against acetylated tubulin to visualize the axonal scaffold. These larvae were treated the same way as in whole mount in situ hybridization procedure. The axonal scaffold was used to align the images to the average reference scaffold and thereby bringing the dapi signal into the reference coordinate system, and this dapi signal image was the starting point of the modeling procedure. First step was to do the intensity correction along the Z-axis. Then the maxima points were estimated, which were further validated by manually checking the overlay of the coordinates with the nuclei. These estimated maxima points were used as the starting point to grow the regions in a way that they meet on the boundary. For the sake of clarity, A describes the procedure for 24 hour larvae. (B) shows the cellular model of larval brain of 48 hour old *Platynereis* larvae. Ba,Bd shows the dapi staining at two different optical sections and Bb, Be shows the corresponding plains of the modeled nuclei. Bc shows the three dimensional reconstruction of the dapi stained embryo used for building the cellular model. Bf shows the size distribution of the nuclei.

2.1.1.7 Features extraction: Expression patterns to cellular expression profiles

One of the main advantages of WMISEP protocol is that the expression pattern genes are brought into a common reference system, which implies that any gene can be compared with any number of other genes. This capability of the protocol leads to a combinatorial explosion of the number of combinations that can be generated, hence making it impossible to analyze all the possible combinations manually. Therefore, to facilitate the analysis of all the data, I developed many methods to automatically extract interesting features from the dataset. Firstly, I developed a cellular model of 48 hpf larval brain in the reference coordinate system. For this I acquired three-dimensional images of nuclei in the larval brain, using a 48 hour larva which was stained with a nuclear stain Dapi, immunostained against acetylated tubulin and treated the same way as a normal gene expression procedure to mimic the shrinking artifacts. Further, I aligned the nuclei channel to the reference brain image, using the axonal scaffold as template, to bring the image into the reference coordinate system. The algorithm and the programs used for building the cellular model of a *Platynereis* 48 hour larval brain are summarized in the Figure 17A and Appendix. As a starting point, the 3D image stacks acquired by Confocal microscopy show a non-uniform intensity distribution along the Z-axis. Therefore, the first step was to correct the intensity along the Z-axis with the assumption that the local mean intensity remains uniform. Subsequently, the local maxima positions were calculated from the nuclear channel image, which were further verified and corrected manually to ensure that they mark the center of the nuclei. These maxima points were

then synchronously grown in 3D space in a grey-weighted manner such that the growing regions meet at the cellular boundary. Afterwards, I investigated the size distribution of the modeled nuclei to assess the quality of the final model. As shown in Figure 17B, it is evident that the size distribution of the modeled nuclei follows a Gaussian distribution, indicating a uniform size distribution. In total 2070 nuclei were modeled for the 48 hour *Platynereis* larval brain. The 3D reconstruction of the original dapi staining used for modeling is shown in Figure 17B-c and the comparison of the model with the original dapi nuclear stained larval brain image at two different depth is shown in Figure 17B-a,b,d,e.

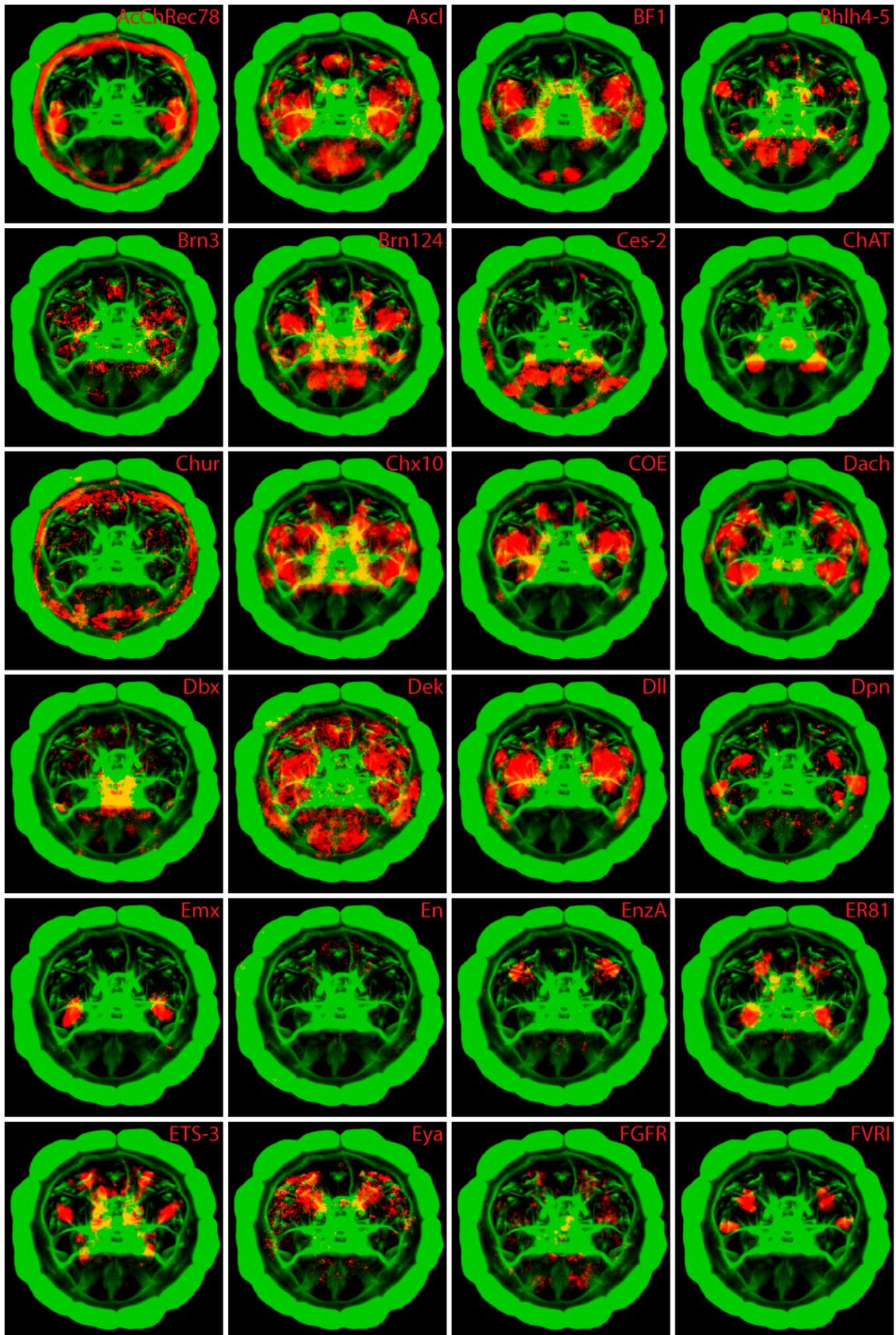
The next step was to generate cellular expression profiles by overlaying the average expression pattern of several genes onto the model. To achieve this, I designed an algorithm for the automatic assignment of the spatial expression of genes to cells in the model simultaneously. Firstly, the average spatial gene expression is superimposed on the cellular model, and then the mean expression intensity value is calculated for each of the cells in the model. Further, the cellular mean expression intensity values were converted into a Z-score which facilitates the decision making if a cell is positive for a given gene. Conceptually, a Z-score value of 1 implies that the value is one standard deviation more than the mean. Generally, a uniform Z-score cutoff of 1 for all the genes already resulted in highly sensitive and specific assignments of the expression to the cells. To further improve the assignment accuracy, I manually optimized the Z-score cutoff for all the genes by manually analyzing the overlay of the model and the expression patterns. Also, I manually validated the final gene expression assignments by overlaying the expression images onto the cellular model. In total I investigated 72 genes (see Table 1) during my PhD work. The average expression patterns for all the genes investigated are shown in the Figure 18. Since the spatial expression of genes is in the same reference system, any combination of genes can be generated for further analysis. The total number of double and triple combinations that can be generated from this data set are 2,556 (${}^{72}C_2$) and 59,640 (${}^{72}C_3$) respectively. Also, 64 genes were successfully fed into the cellular model to generate expression profiles for all of the 2070 cells in *Platynereis* 48 hour old larval brain, resulting in a table of 2071 rows and 65 columns, which were then used for several clustering analysis (see Section 2.1.2.2 and 2.1.2.3)

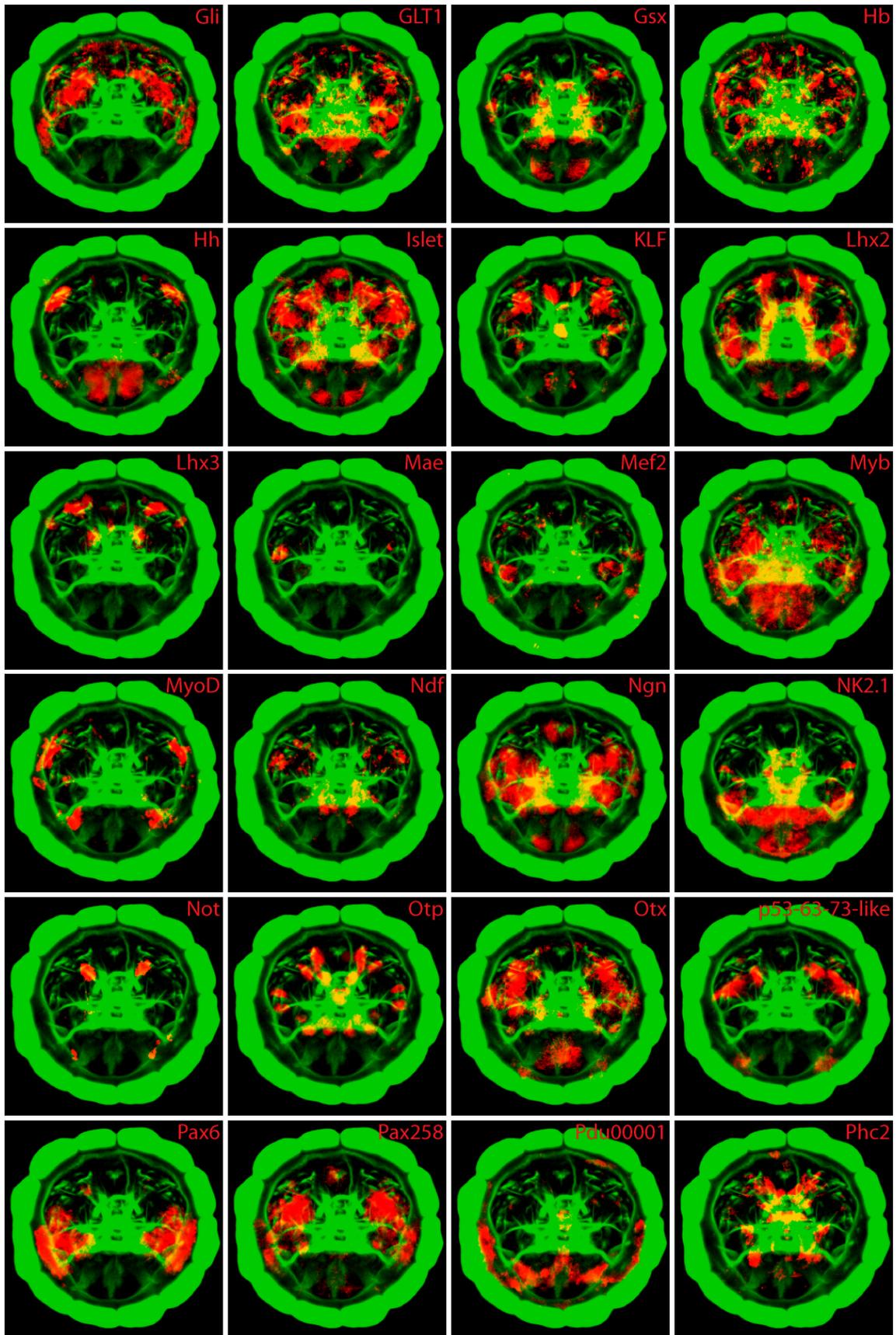
Table 1: List of genes investigated in this thesis

S. No.	Symbol	Description	Source	Images Acquired By
1	AcChRec78	Acetyl Choline Receptor	IB0AAA19CA10EM1	Raju Tomer
2	Ascl	achaete-scute	IB0AAA28BE11EM1	Raju Tomer
3	BF1	Forkhead box protein G1, Brain Factor1	IB0AAA24AG07EM1, B. Prudhomme	Raju Tomer
4	BHLH4-5	Basic-helix-loop-helix	IB0AAA17CG03EM1	Raju Tomer
5	Brn124	homologue of Brain-specific homeobox/POU domain protein 1, 2, 4	IB0AAA33AC08EM1	Raju Tomer
6	Brn3	homologue of Brain-specific homeobox/POU domain protein 3	K. Tessmar-Raible	Raju Tomer
7	ces-2	cell-death specification protein 2	IB0AAA25DF03EM1	Raju Tomer
8	ChAT	Choline Acetyltransferase	(Denes et al., 2007)	Nicola Kegel
9	Chur	Churchill	IB0AAA20CH08EM1	Raju Tomer
10	Chx10	homeodomain protein	(Denes et al., 2007)	Raju Tomer
11	COE	Collier/Olf1/EBF (COE)	IB0AAA42CB09EM2	Raju Tomer
12	Dach	Dachshund	Raju Tomer	Raju Tomer
13	Dbx	Homeobox gene	(Denes et al., 2007)	Raju Tomer
14	DEK	Protein DEK, May have a function in the nucleus	IB0AAA15CC12EM1	Raju Tomer
15	Dll	Distal-less	B. Prudhomme	Raju Tomer
16	Dpn	Deadpan	IB0AAA31CC01EM1	Raju Tomer
17	Emx	Empty spiracles homolog, Homeobox protein	K. Tessmar-Raible	Raju Tomer
18	Engrailed	Homeobox gene	(Prud'homme et al., 2003)	Raju Tomer
19	EnzA	sepiapterin synthase A	PhD Thesis - Keren Guy	Keren Guy
20	ER81	ETS translocation variant 1, ETV1	IB0AAA20BB09EM1	Raju Tomer
21	ETS3	DNA-binding protein D-ETS-3	IB0AAD5YG04CM1	Raju Tomer
22	Eya	Eyes Absent	RACE product by Keren Guy	Raju Tomer
23	FGFR	FGF Receptor	Patrick Steinmetz	Raju Tomer
24	FVRI	FVRlamide peptide	Jekely et. al. 2008	Raju Tomer
25	Gli	GLI-Kruppel transcription factor	Fay Christodoulou, Kristin Tessmar-Raible	Keren Guy
26	GLT1	Glutamate transporter	AS Denes	Raju Tomer
27	Gsx	ParaHox-GSX	(Denes et al., 2007)	Raju Tomer
28	Hb	Hunchback, belongs to Zinc finger protein family	(Kerner et al., 2006)	Raju Tomer
29	Hh	Hedgehog	Kristin Tessmar-Raible	Raju Tomer
30	Islet	Insulin gene enhancer protein ISL	IB0AAA30CH08EM1	Raju Tomer
31	KLF	Krueppel-like factor 4	IB0AAA32BG02EM1	Raju Tomer
32	Lhx2	LIM/homeobox protein 2	IB0AAA31DE11EM1, (Denes et al., 2007)	Raju Tomer

33	Lhx3	LIM/homeobox protein 3	Kristin Tessmar-Raible	Raju Tomer
34	Mae	ETS activity modulator	IB0AAA35DD02EM1	Raju Tomer
35	Mef2	Myocyte-specific enhancer factor 2	Patrick Steinmetz	Raju Tomer
36	Myb	Myb proto-oncogene protein	IB0AAD11YG06CM1	Raju Tomer
37	MyoD	Myoblast determination protein	IB0AAA28CH12EM1	Raju Tomer
38	Ndf	neurogenic differentiation	IB0AAA27DH04EM1	Raju Tomer
39	Ngn	Neurogenin	IB0AAA15BG11EM1	Raju Tomer
40	NK2.1	Homeodomain protein	(Tessmar-Raible et al., 2007)	Raju Tomer
41	Not	homeodomain protein	Detlev Arendt	Nicola Kegel
42	Otp	Homeobox protein orthopedia	(Tessmar-Raible et al., 2007)	Raju Tomer
43	Otx	Orthodenticle homolog	(Arendt et al., 2001)	Raju Tomer
44	p53-p63-p73-like	tumor protein	IB0AAA34AG01EM1	Raju Tomer
45	Pax258	Homologue of Paired box homeotic gene 2, 5, 8	(Denes et al., 2007)	Raju Tomer
46	Pax6	Paired box homeotic gene 6	(Arendt et al., 2002)	Raju Tomer
47	Pdu00001	Platynereis specific gene 1	Pdu_48_1_B02	Raju Tomer
48	Phc2	Prohormone convertase 2	IB0AAA27AB10EM1	Nicola Kegel
49	Prox1	prospero homeobox 1	PhD Thesis - Keren Guy	Keren Guy
50	REQU-DPF	Zinc finger protein	IB0AAA19CF09EM1	Raju Tomer
51	ROpsin	Rhabdomic opsin	(Arendt et al., 2004)	Raju Tomer
52	Rx	Retina and anterior neural fold homeobox protein	(Arendt et al., 2004)	Raju Tomer
53	SGT1	Ecdysoneless homolog	IB0AAA37AB06EM1	Raju Tomer
54	Sim	Single-minded homolog	(Denes et al., 2007)	Raju Tomer
55	Six12	Sine oculis homeobox homolog 1, 2	(Arendt et al., 2002)	Raju Tomer
56	Six3	Sine oculis homeobox homolog 3	D. Arendt	Raju Tomer
57	SMAD2-3	Mothers against decapentaplegic homolog 2, 3	IB0AAD22YI07CM1	Raju Tomer
58	Smr	SANT domain protein SMRTER	IB0AAA32AG02EM1	Raju Tomer
59	Svp	Nuclear receptor subfamily 2 group F member 3, isoforms B/C	Raju Tomer	Raju Tomer
60	Syt	Synaptotagmin	(Tessmar-Raible et al., 2007)	Raju Tomer
61	Tll	tailless, Nuclear receptor subfamily 2 group E member 2	Detlev Arendt	Raju Tomer
62	TrpHyd	tryptophane hydroxylase	(Denes et al., 2007)	Raju Tomer
63	Tryp2,3	tryptophan 2,3-dioxygenase	PhD Thesis - Keren Guy	Keren Guy
64	VACht	Vesicular Acetylcholine Transporter	(Denes et al., 2007)	Raju Tomer
65	VGLUT	Vesicular Glutamate Transporter	Alexendru Denes	Raju Tomer
66	Wnt5	Secreted glycoprotein belonging to wingless (wg)/Wnt family	IB0AAA23BF01EM1, Guillaume Balavoine	Raju Tomer
67	Wnt8	Secreted glycoprotein belonging to wingless (wg)/Wnt family	IB0AAA36CA12EM1, Guillaume Balavoine	Raju Tomer

68	WntA	Secreted glycoprotein belonging to wingless (wg)/Wnt family	Guillaume Balavoine	Raju Tomer
69	Xbp1	X box-binding protein 1	IB0AAA33BF04EM1	Raju Tomer
70	ZBT24	Zinc finger and BTB domain-containing protein 24	IB0AAA42AC11EM1	Raju Tomer
71	ZFAT-like	Zinc finger protein	IB0AAA37DA10EM1	Raju Tomer
72	Zic	Zinc finger protein of the cerebellum	IB0AAA34BD03EM1	Raju Tomer





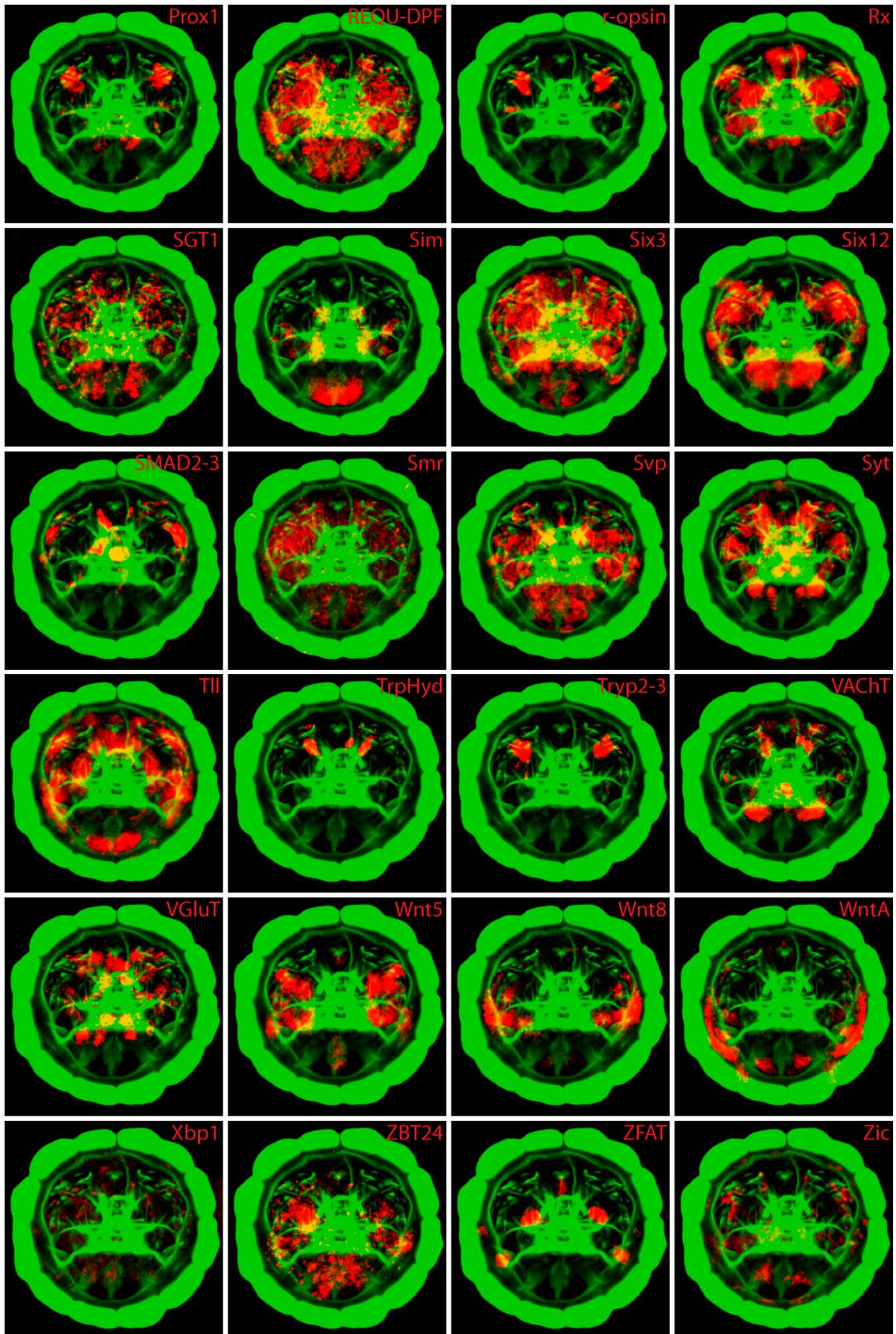


Figure 18. Summary of the average expression patterns of all the genes investigated in 48 hour old larval brain.

Green shows the average reference axonal scaffold of 48 hour old *Platynereis* larval brain and red is the average expression pattern of the gene (labeled in red) after WMISEP protocol. All the pictures are maximum Z-projections. All are apical views. Dorsal is up and ventral is down.

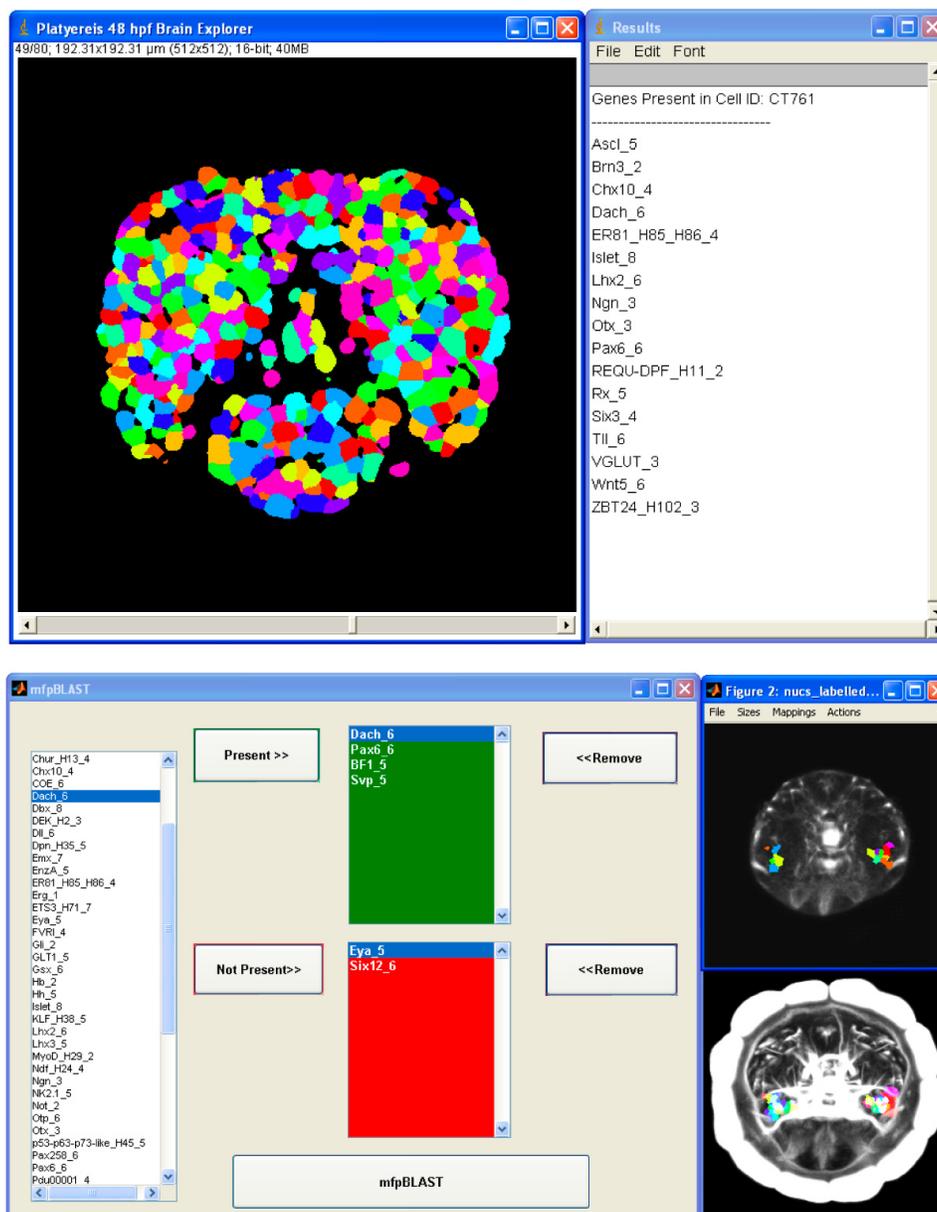


Figure 19. Interfaces for searching the database.

(Top) Screenshot of BrainExplorer tool. This tool is implemented as a plugin in ImageJ and allows clicking on any cell to visualize the list of genes expressed in it. (Bottom) shows the screenshot of mfpBLAST (molecular fingerprint BLAST), which is implemented as graphic user interface in MATLAB. This tool allows searching for cells which shows expression. This interface allows searching the cellular brain model for cells that express a specific combination of genes but do not express another specific combination of

genes. As an output mfpBLAST displays an animated stack showing the positive cells superimposed on the average reference brain image. For instance, search query and result for the molecular fingerprint of mushroom body anlagen cells is shown in the snapshot. The source codes of both the tools are presented in Appendix.

2.1.1.8 Search interfaces for the database of cellular expression profiles

Having established a detailed cellular expression profile database, it was essential to create interfaces which will allow querying the database in a flexible manner. Therefore, I developed a couple of tools for searching the database either starting from the combination of genes query or from the cellular model to find out the genes expressed in a particular cell. Firstly, I implemented a tool, called BrainExplorer (Figure 19-top), as a plugin in ImageJ (<http://rsb.info.nih.gov/ij/>). This interface visualizes the *Platynereis* larval brain model in a graphic window stack, with a possibility of going up and down in the Z- direction by moving the slider. On clicking on a particular cell, the program searches the database for the entry for that cell and extracts the list of genes expressed in it, which are then displayed as a list in a separate window. Secondly, I developed mfpBLAST (Molecular Fingerprint BLAST) (Figure 19-bottom) as a graphic user interface in MATLAB. As the name suggests, this interface allows searching the cellular brain model for cells that expresses a specific combination of genes but do not express another specific combination of genes. As an output mfpBLAST displays an animated stack showing the positive cells superimposed on the average reference brain image. For instance, (Figure 19-bottom) shows a search query and result for the molecular fingerprint of mushroom body anlagen cells. The source codes of both the tools are presented in the Appendix.

*2.1.1.9 Extending WMISEP to other stages of development of *Platynereis dumerilii**

After establishing the WMISEP protocol for 48 hpf *Platynereis* larval brain, the next step was to optimize this protocol for other temporal stages of *Platynereis* larval development. This will present the opportunity to investigate the dynamics of co-expression of several genes and predict the cell type specification gene regulatory networks. Therefore, I aimed to optimize the protocol for two other temporal stages: 56 hour old larvae and 4 day old

young worm. The procedures established are generally the same as described above for 48 hpf larvae but with few differences. Briefly, I used immunostaining against acetylated-tubulin as the template channel to align images of different individuals for generating spatial expressions patterns of several genes. Further, I generated average reference brains for 56 hpf larvae using three-dimensional axonal scaffold images acquired from 27 different individuals and for 4 dpf using 14 different individuals using similar procedures as described above. The average reference images are shown in Figure 20 for 56 hpf and in Figure 23A-C for 4 dpf. It is evident that the average reference image has retained majority of the features (for instance, see legend of Figure 20 and cyan arrows in Figure 23A-C), suggesting high sensitivity. Furthermore, I conducted proof of principle studies for 56 hpf (Figure 21). It is apparent that the sensitivity achieved in 56 hpf is as good as that achieved for 48 hpf. The WMISEP protocol for 4 dpf is still a work in progress and requires further experiments and optimizations. Subsequently, I established methods for averaging the expression patterns of same gene but from different individuals. Afterward I generated average expression information for several genes for 56 hpf including *Ascl*, *BF1*, *Dach*, *Dbx*, *Dll*, *Emx*, *Ngn*, *Pax6*, *Six3*, *Syt*, *VACHT* and *VGluT* (shown in Figure 22A-L) and for 4dpf (preliminary) including *BF1*, *Dach*, *Otx*, *Pax6*, *Syp*, *VACHT* and *VGluT* (shown in Figure 23 and 24). Similar to 48 hpf expression images, combinations of any number of different genes can be generated from this data source.

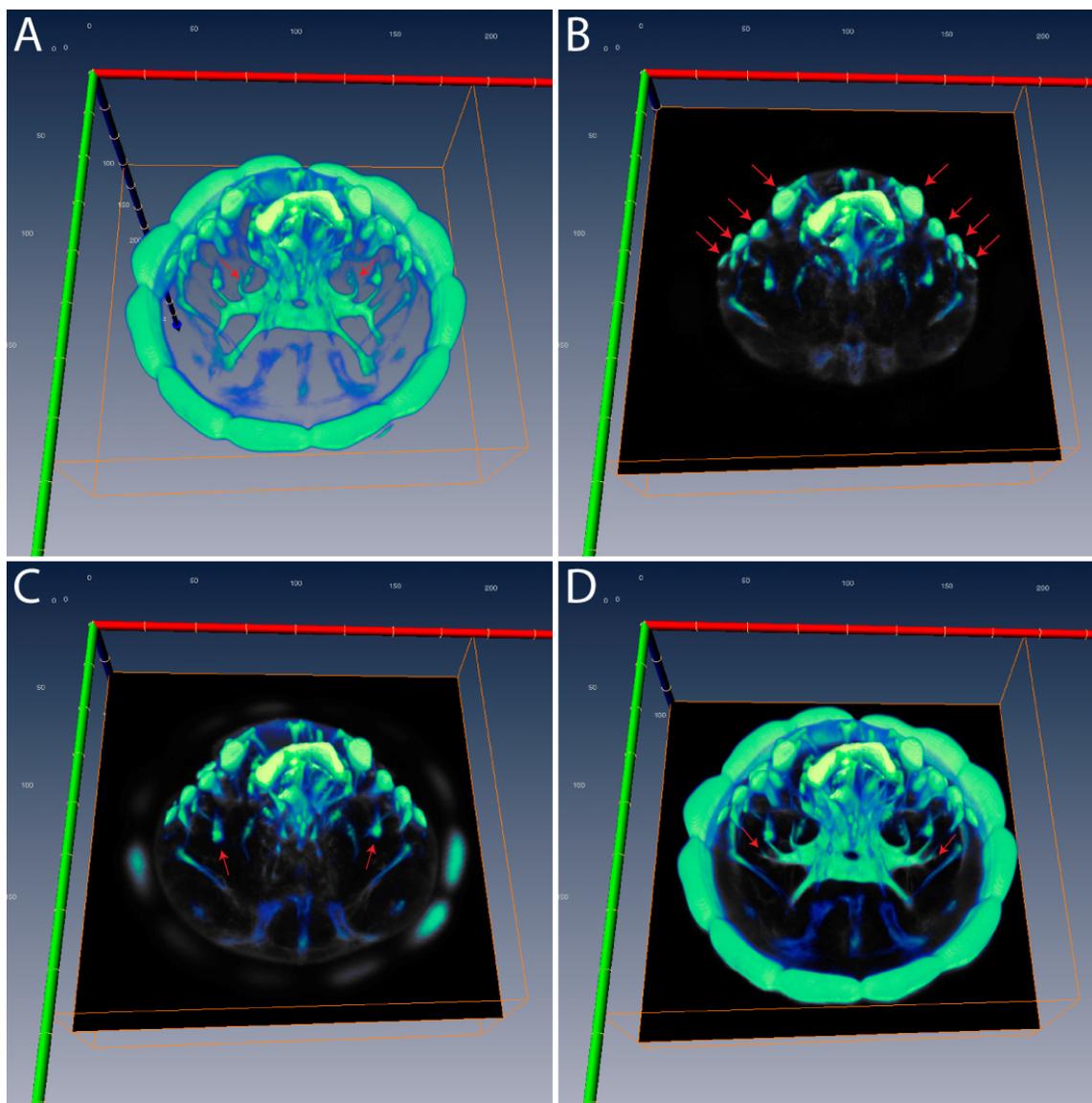


Figure 20. Three-dimensional reconstruction of the average reference axonal scaffold of 56 hour old *Platynereis* larval brain.

(A) The three dimensional reconstruction of the average reference axonal scaffold was generated using Amira (<http://www.amiravis.com/>). (B-D) illustrates the fine details of the average reference image using a black slider at various optical depths. It is clear that the average reference larval axonal scaffold is highly symmetric and has retained majority of the neuronal features. For instance, red arrows in A, B, C and D points towards the symmetrically located connectives, sensory organs, superficial sensory synapses and an unknown connective respectively.

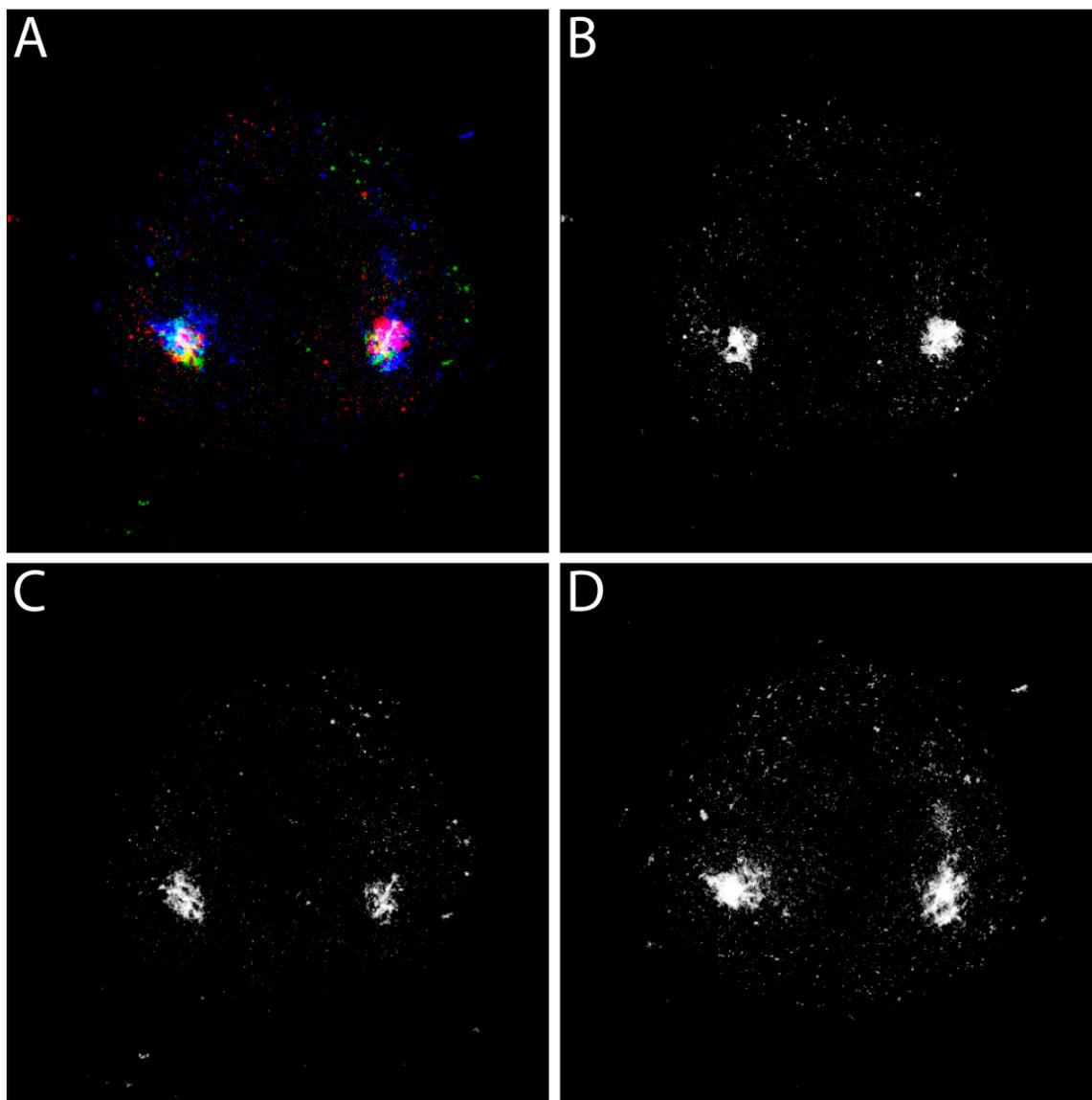


Figure 21. Proof of principle experiment of WMISEP protocol for 56 hpf.

Confocal images were acquired from three different individual larvae stained for the same gene, *Emx*. The images were then aligned using WMISEP. B-D are the maximum z-projections of three individual images and A is the maximum z-projection of the three-color (RGB) merge of these three images. It is clear that the expression signals overlap very well, in accordance with the expectation.

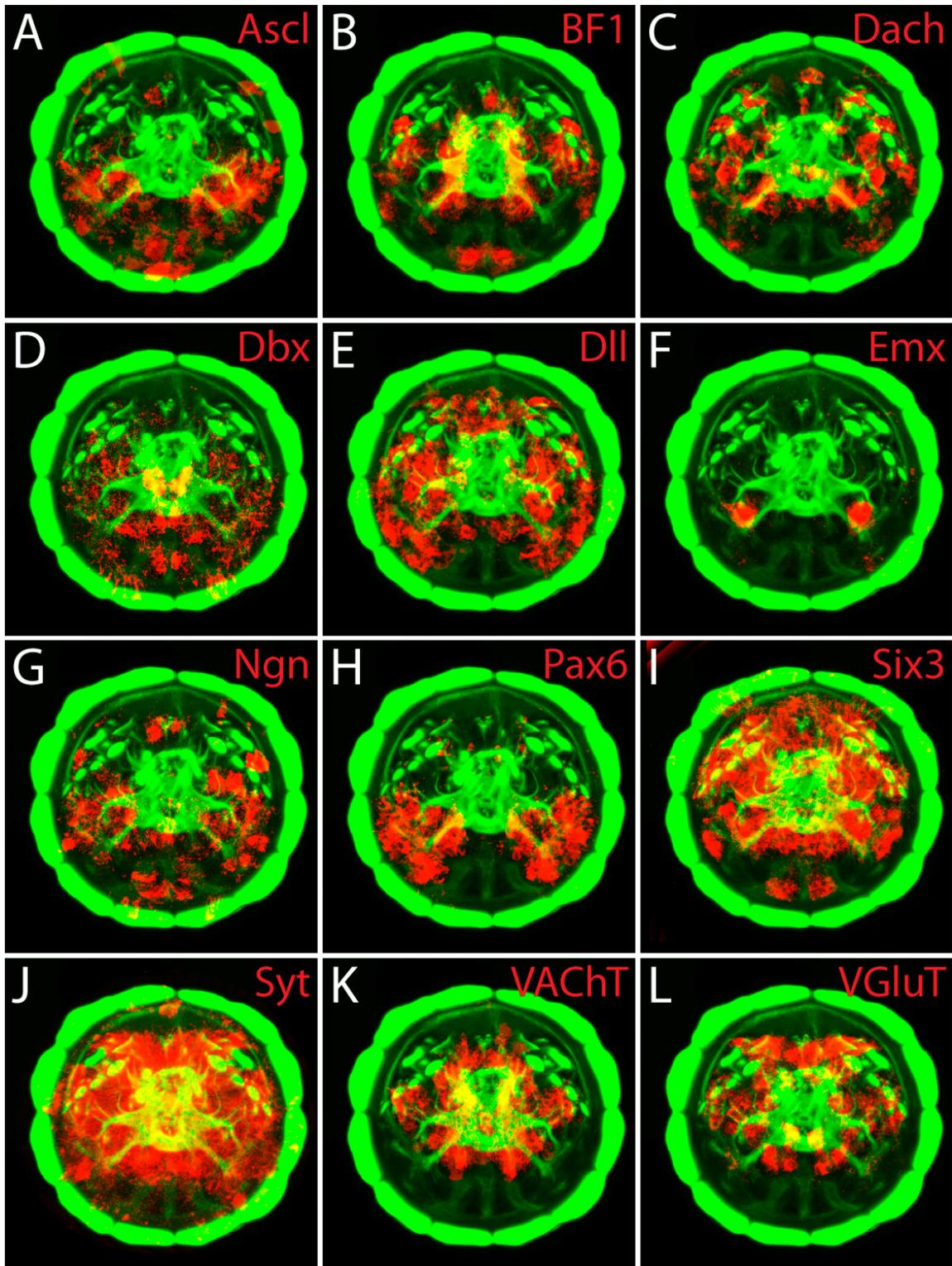


Figure 22. Collection of average expression patterns of genes investigated in 56 hour old larvae.

Green shows the average reference axonal scaffold of 56 hour old *Platynereis* larval brain and red is the average expression pattern of gene (labeled in red) after WMISEP protocol. All the pictures are maximum Z-projections. All are apical views. Dorsal is up and ventral is down.

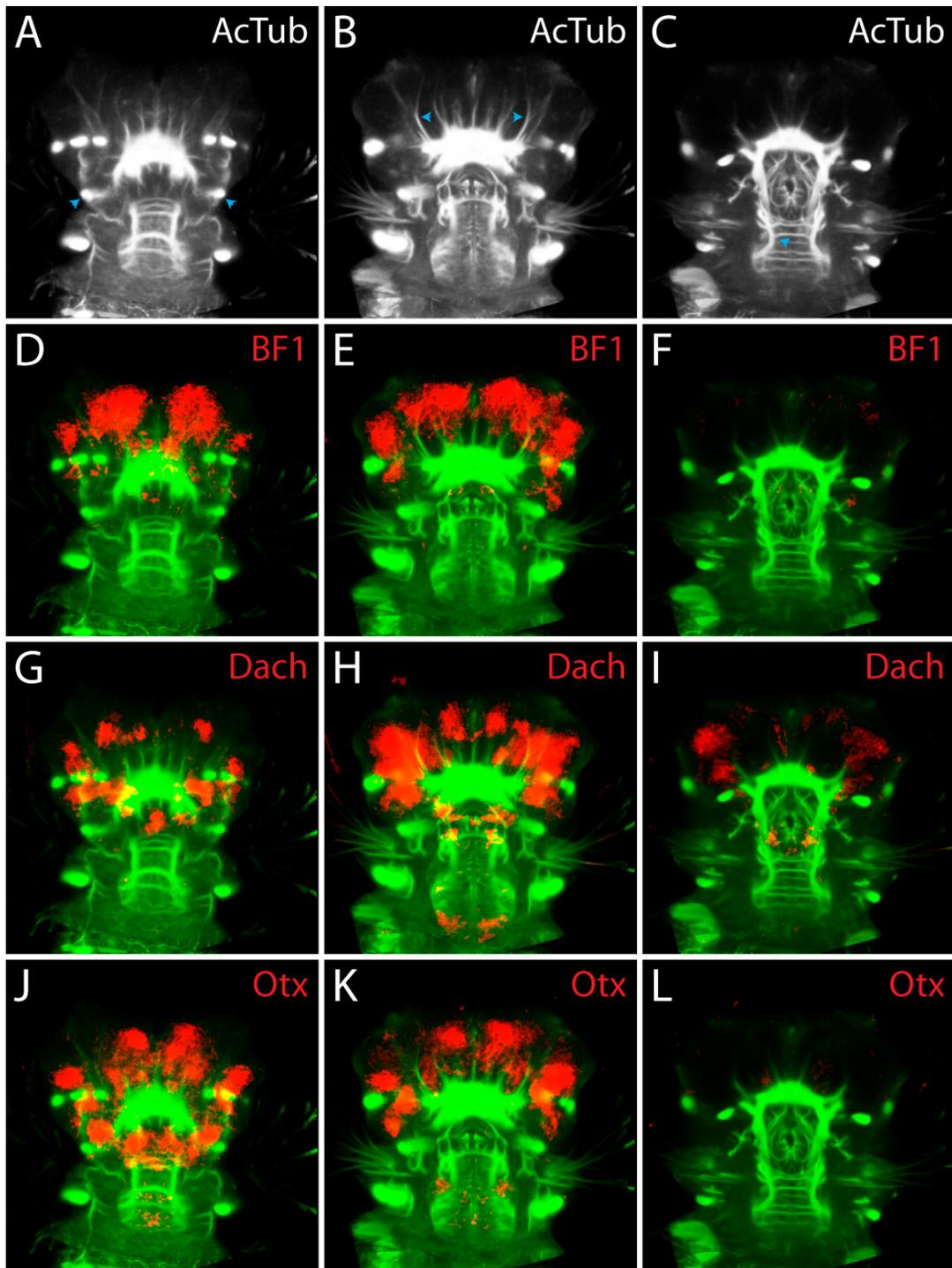


Figure 23. Collection I of average expression patterns of genes investigated in 4 days old *Platynereis* Nechtochaete.

(A-C) Maximum z-projections of the average reference axonal scaffold of 4 days old nechtochaete at three different optical depths in dorsal to medial direction. Anterior is up and posterior down. (D-L) Green shows the average reference axonal scaffold of 4 days old *Platynereis* nachtochaete and red is the average expression pattern of gene (labeled in red) after WMISEP protocol. All the pictures are maximum Z-projections at three optical depths in dorsal to ventral orientation. Anterior is up and posterior down.

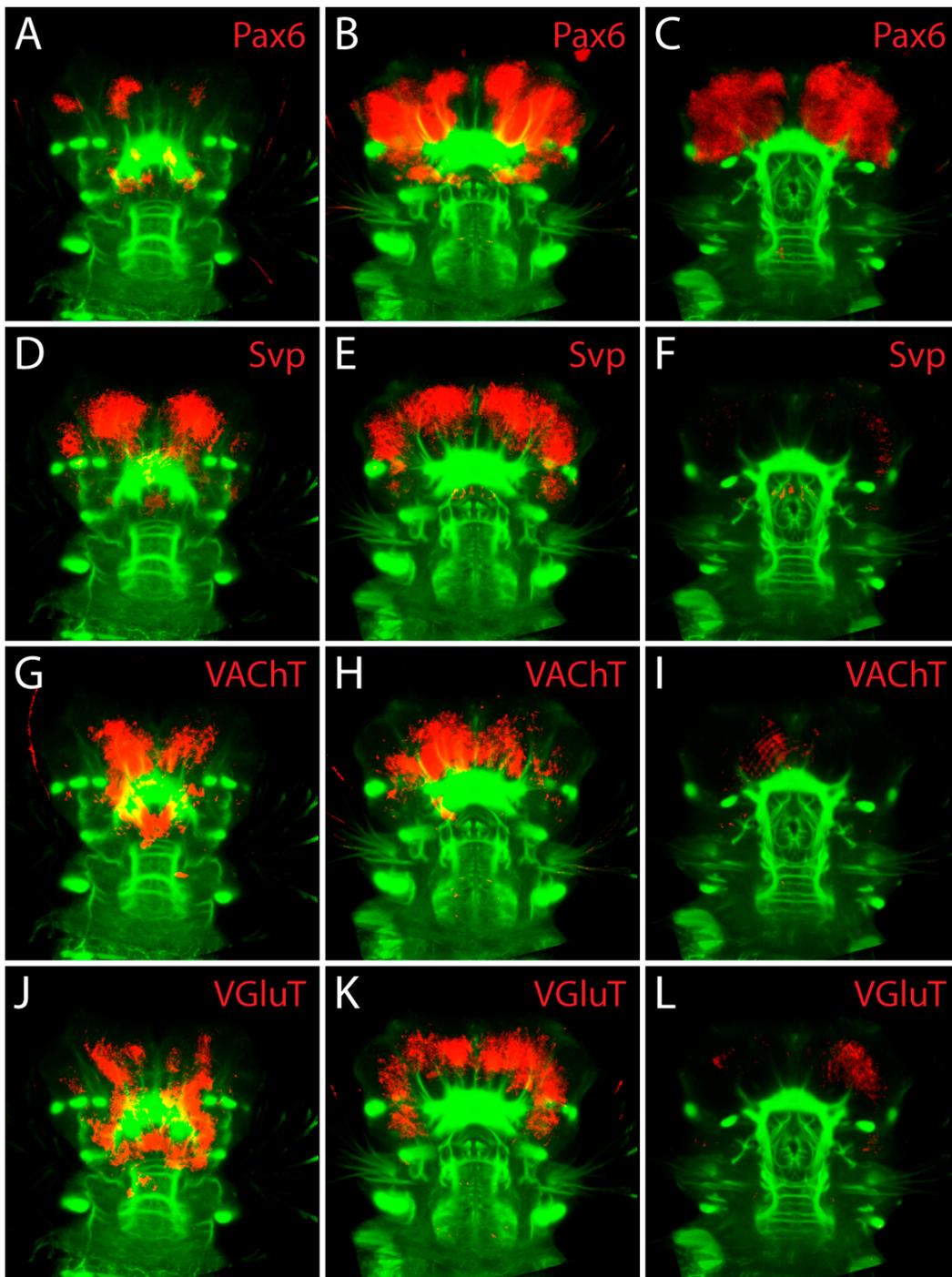


Figure 24. Collection II of average expression patterns of genes investigated in 4 days old *Platynereis* Nechtochaete.

Anterior is up and posterior down. Green shows the average reference axonal scaffold of 4 days old *Platynereis* nachtochaete and red is the average expression pattern of gene (labeled in red) after WMISEP protocol. All the pictures are maximum Z-projections at three optical depths in dorsal to ventral orientation. Anterior is up and posterior down.

2.1.2 Clustering analysis

2.1.2.1 Clustering approaches

Hierarchical clustering and data partitioning are the two most commonly used clustering techniques (D'Haeseleer, 2005). Hierarchical clustering methods work by subdividing each cluster into smaller clusters iteratively and represent the resulting clusters as dendograms. The two most important decisions to be made for using hierarchical clustering are: how is the similarity measure defined and how is the intercluster distances (linkage) defined. The most commonly used similarity measures are Euclidean distance, Centered correlation, Uncentered correlation and City-block (D'Haeseleer, 2005). The intercluster distances are defined as the linkage functions and the most commonly used linkage functions are: single linkage (the shortest distance between the two clusters), complete linkage (largest distance between two clusters) and average linkage (average distance between the members of two clusters). For Hierarchical clustering, the uncentered correlation as the similarity measure and the complete linkage as intercluster distance measure work better in most of the cases (D'Haeseleer, 2005). On the other hand, data partitioning approaches work by subdividing the data into a specified number of clusters. The main disadvantages of this method are that firstly it needs to know the number of clusters present in the data and secondly it does not imply any hierarchical relationship between the clusters. However, recently a hybrid of the hierarchical and data partitioning approaches was reported – HOPACH (Hierarchical Ordered Partitioning And Collapsing Hybrid)(J. van der Laan and Pollard, 2003). This method combines the benefits of both the hierarchical clustering and the data partitioning techniques and works by building hierarchical relationships of the clusters. It recursively partition the data while ordering and collapsing the clusters at each level. It identifies the number of maximally homogeneous clusters in the data by optimizing MSS (Mean/Median Split Silhouette) criteria. Furthermore, the hopach package provides tools for non-parametric bootstrapping to estimate the probability of each element belonging to each of the clusters.

A. The average reference image of larval brain is shown in green and the cells belonging to corresponding sub-clusters are shown in red. (C) shows the representative genes (mean) defining the molecular fingerprint of sub-clusters marked in A. The intensity of red color in the bars corresponds to the percentage of cells positive for that gene in that particular cluster. The dendrogram shows the relatedness of the genes in terms of their expression patterns.

2.1.2.2 Clustering of *Platynereis* larval brain cells based on gene expression profile

I used several clustering approaches to reveal the interrelationships of *Platynereis* larval brain cells, based on their gene expression profile. Firstly, I used a Hierarchical clustering approach, with uncentered correlation as the similarity measure and complete linkage as the intercluster distance measure, to cluster the cellular gene expression profiles. The dataset used for the clustering analysis contained manually validated expression information for 64 genes (Figure 18) in 1792 cells (after filtering the dataset of 2070 cells with the criteria that at least 2 genes are expressed) of *Platynereis* larval brain. The overall clustering results are shown in Figure 25. It is apparent that signatures for many meaningful clusters can be identified. Some of the interesting clusters are highlighted with unique colors and a label (Figure 25A). Furthermore, I developed methods to visualize the spatial localization of the cells, belonging to a particular cluster, in the larval brain model. For instance, Figure 25B shows the spatial organization of the cells belonging to 9 distinct clusters. It is evident that the cells in a given cluster tend to be symmetrically distributed spatially and are compactly organized in space. Furthermore, I visualized the representative genes for each of the clusters, which most likely represent the molecular fingerprint of that particular cluster type. Figure 25C shows the defining genes for the clusters shown in Figure 25A. By analyzing the markers in the representative genes of the clusters, specific identities can be assigned to the cluster types. For instance, Cluster1 expresses *Gli* and therefore can be defined as the cells receiving Hedgehog signals; Cluster 2 expresses *BF1*, *Lhx2* and *Gsx*, homologous to vertebrate ventral telencephalon (see Section 2.3.1 for details); Cluster3 expresses *r-opsin* and therefore includes the adult eye cells; Cluster4 expresses *Dach*, *Six12*, *Eya*(partially) and *Six3*, and likely includes the eye field cells; Cluster5 expresses *Dach*, *Pax6*, *BF1*, *Lhx2*, *Wnt5* and *Syp* but not *Eya* and *Six12*, and therefore includes the mushroom body cells (see section 2.2.4); Cluster6 expresses *Rx*, *Dll*, *Ngn*, *Ascl* and *Six3*, and includes

cells of yet unknown identity; Cluster7 expresses *BF1*, *Lhx2*, *Gsx* and *NK2.1*, and therefore includes cells molecularly similar to Cluster2.

It is important to investigate the statistical significance of these clusters and the relationships among clusters. For this I used another approach called HOPACH (Hierarchical Ordered Partitioning And Collapsing Hybrid) for clustering the same dataset, using cosangle (same as uncentered correlation) as the similarity measure. As depicted in Figure 26, this analysis yields three super clusters of the *Platynereis* larval brain cells, based on their expression profile. Besides it is evident that the relationships between the clusters could be better understood with this method than the Hierarchical clustering method alone. Also, to judge the statistical significance of the clusters, I used bootstrap re-sampling analysis. Figure 27 shows the proportional reproducibility of each cell into all the three clusters after 1000 bootstrap re-sampling. Cluster A has a mean reproducibility proportion value of 0.8391 with a standard deviation value of 0.1642, Cluster B has a mean reproducibility value of 0.755 with the standard deviation value of 0.1701 and Cluster C has a mean reproducibility value of 0.7766 with the standard deviation value of 0.1852. Furthermore, I visualized the spatial location of the cells belonging to the three clusters in the *Platynereis* larval brain model. As shown in a series of optical sections in Apical to Basal orientation in Figure 28, it is evident that Cluster A mainly contains lateral cells, Cluster B mainly contains medial cells and Cluster C contains cells in the stomadeum. Moreover, I analyzed the mediods of the three clusters, representing the defining genes of the clusters shown in Figure 26. As evident in Figure 27C, many of the genes used for the analysis are present in mediods of all the three clusters, with some cluster-specific over-represented genes. Finally, the three super-clusters are further divided into many sub-clusters, which may represent distinct cell types.

Additionally, the clustering approaches could also be used to study various cell types emerging from the expression domain of a particular master-regulator gene. For instance, I investigated the cell types that emerge from *Dach* expression domain. For this, I selected all the cells from the dataset which were positive for *Dach* (333 out of 2070 cells in *Platynereis* larval brain) and clustered the subset using Hopach clustering approach, in a similar way as mentioned above. As shown in the Figure 29, it is apparent that there are

three major clusters of cells emerging from *Dach* expressing cells. Then I carried out the bootstrap analysis to calculate the significance of these clusters, which resulted in mean proportion reproducibility values of 0.729268 with a standard deviation of 0.190 for Cluster 1, 0.827995 with a standard deviation of 0.1833 for Cluster 2 and 0.757341 with a standard deviation of 0.1841 for Cluster 3. Furthermore, I analyzed the spatial localization of the cells belonging to the three clusters. Looking at the representative genes of the clusters, it is evident that Cluster2 co-expresses *BF1*, *Wnt5*, *Svp*, *Rx*, *Tll*, *Lhx2* and *Pax6* with *Dach*, which resembles the molecular fingerprint of mushroom body cell types (see Section 2.2.4) and therefore represents mushroom body anlagen cells.

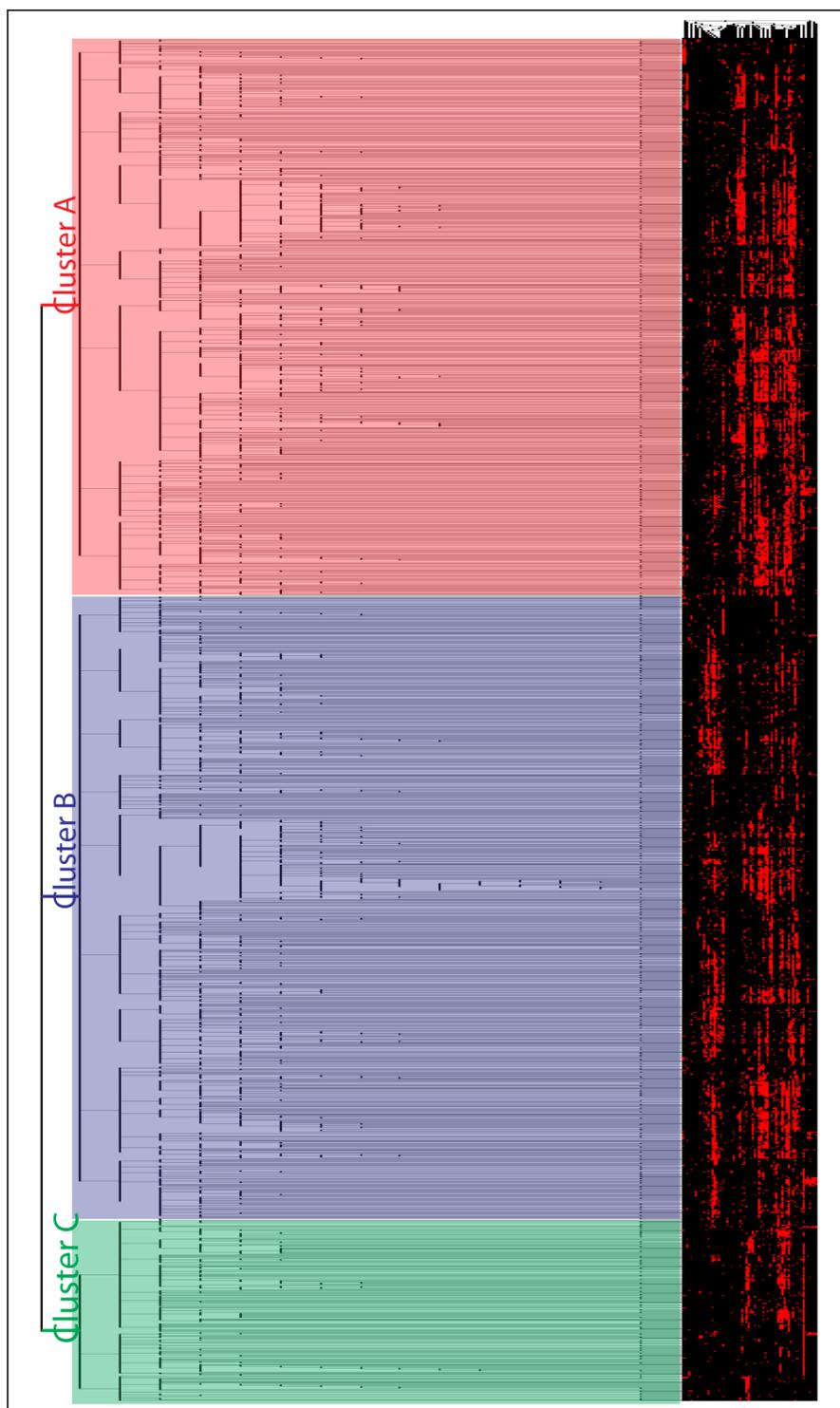


Figure 26. Dendrogram representation of hopach clustering results of all the cells in 48 hour old *Platynereis* larval brain.

There are three super clusters of cells in *Platynereis* larval brain, marked by cluster A (red), cluster B(blue) and cluster C(green). The cells are represented as horizontal lines (in the right end of the dendograms) and the genes are represented as vertical lines. Red pixels indicate the presence and black pixels indicate the

absence. The super-clusters are further divided into sub-clusters. The clustering was done with hopach package in R statistical computing language.

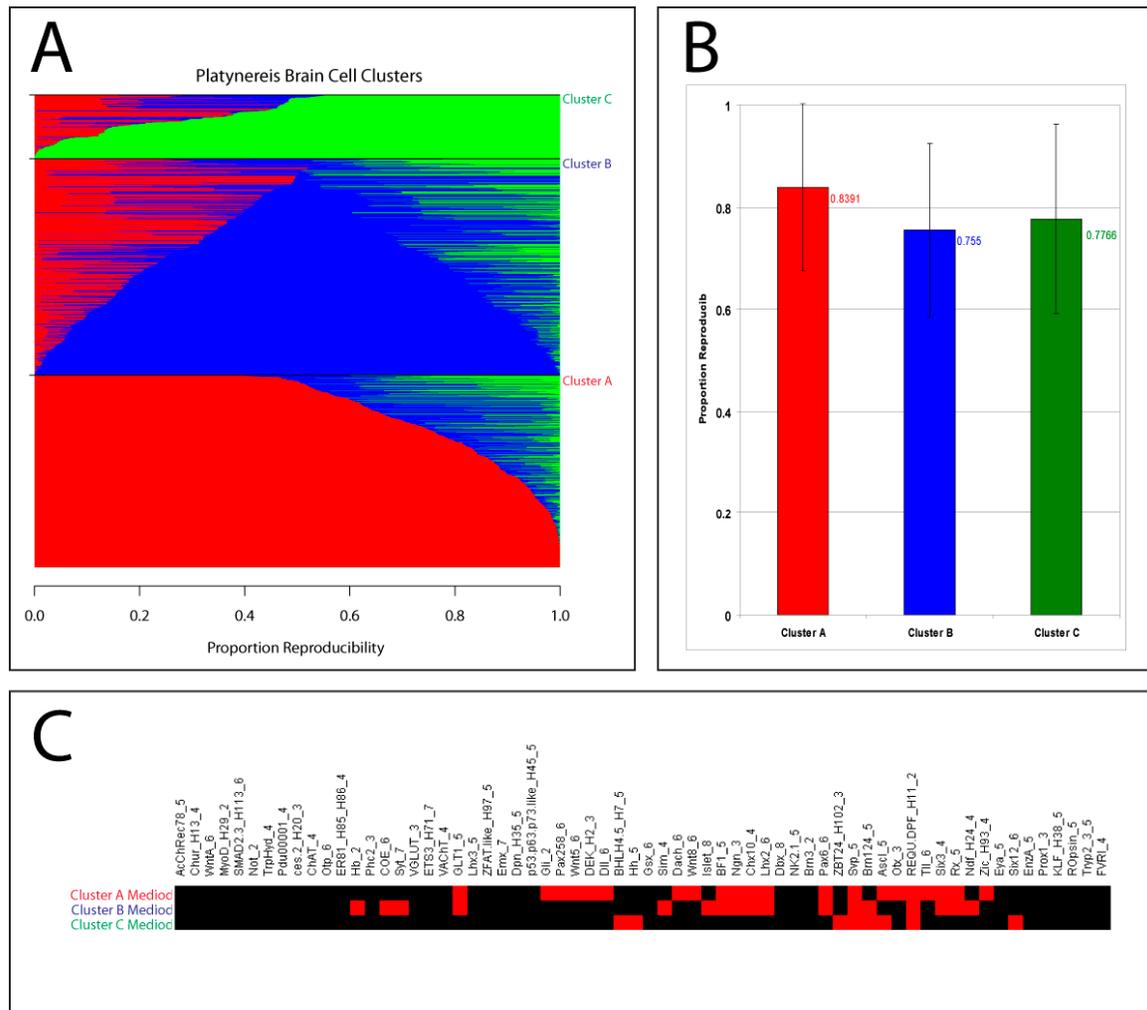


Figure 27. Statistical significance and mediods of the three super clusters of larval brain cells as revealed by hopach analysis.

(A) shows the proportion reproducibility of cells in the three super clusters shown in Figure 26. A horizontal line represents a cell and the proportional coloring represents the proportion reproducibility in cluster of that color. The graph is divided into three areas (marked by cluster A, B and C) for three distinct clusters. Bottom to Top shows ClusterA (red), ClusterB (blue) and ClusterC (green). (B) summarizes the proportion reproducibility graph shown in A. The error bars mark the standard deviation of the proportion reproducibility values. Cells in cluster A, B and C have mean reproducibility values of 0.8391, 0.755 and 0.7766 respectively. (C) shows the mediods of the three super clusters in multi-dimensional gene space. Row represents the coordinates of the mediods and column represents genes. Red square means a gene is present in the mediod of that particular cluster. The gene names end with a number, which represents the number of images of expression pattern that were used for generating the average expression pattern. For instance, *Dach*₆ means 6 different individual larvae were scanned for generating the average expression pattern for *Dach*.

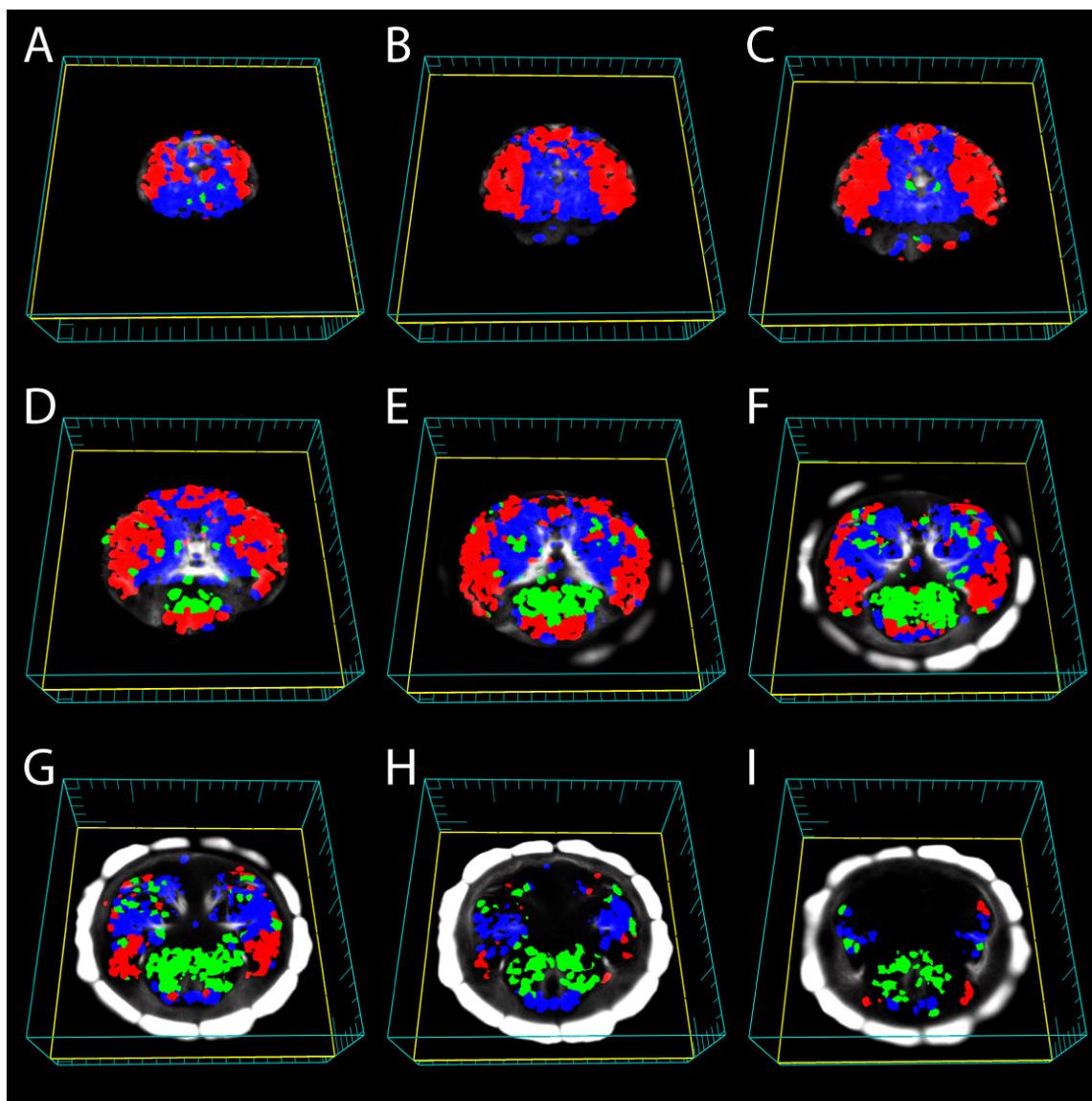


Figure 28. Spatial localization of the larval brain cells belonging to the three super clusters revealed by hopach analysis.

A-I shows a series of optical sections, of cellular model of *Platynereis* 48 hour old larval brain, in apical to basal orientation. The coloring of cells corresponds to the color of super clusters (shown in Figure 26) they belong to. The cells belonging to cluster A are marked red, cells belonging to cluster B are marked blue and cells belonging to cluster C are marked green. The reference average axonal scaffold is shown in white. The major tick marks in the frame represent 50 microns. It is apparent that the cells belonging to cluster A are mainly lateral, cells belonging to cluster B are medial and cells belonging to cluster C are present in stomaedeum.

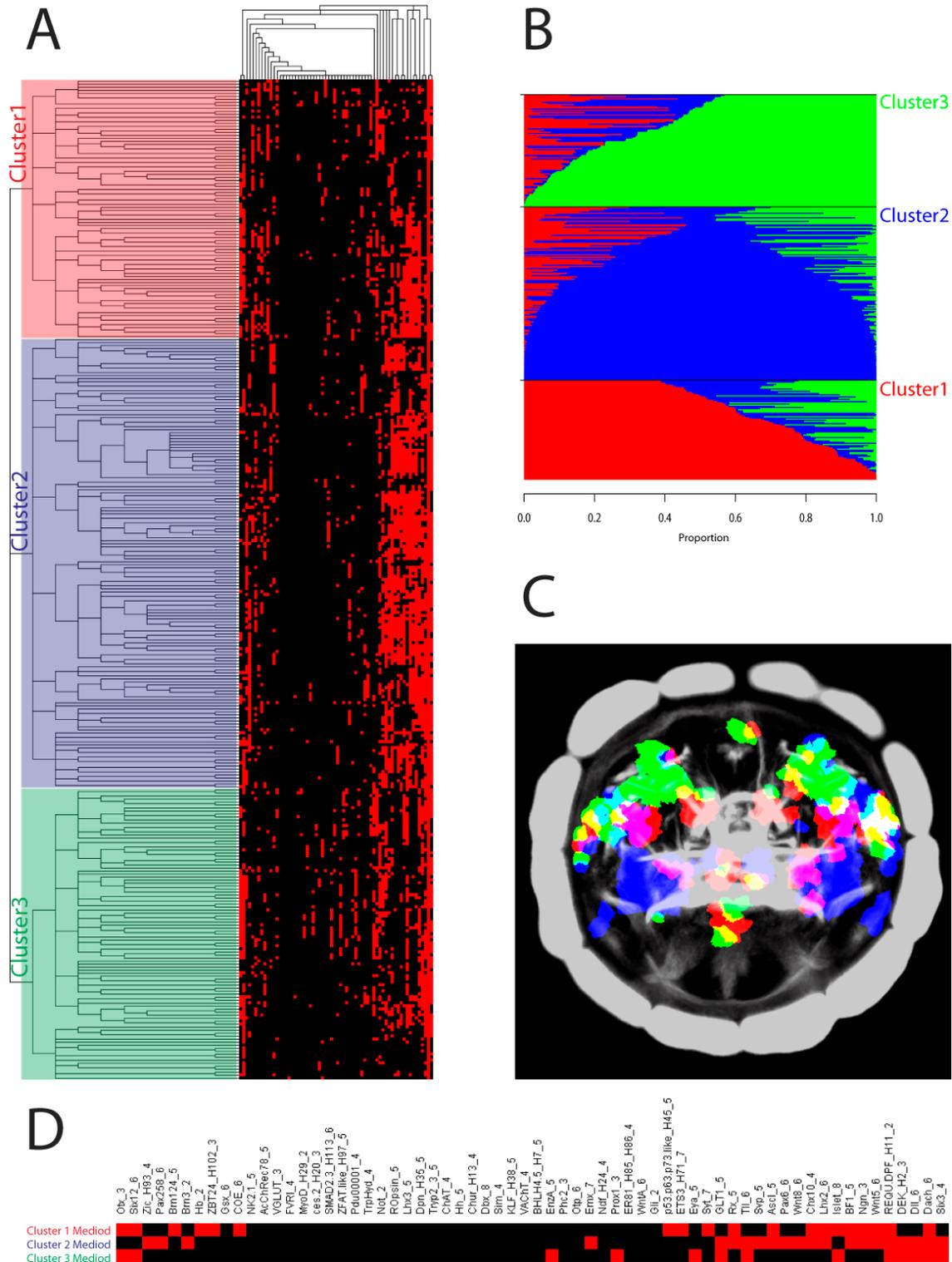


Figure 29. Dendrogram representation of hopach clustering results of all *Dach* positive cells in 48 hour old *Platynereis* larval brain.

(A) There are three super clusters of all the *Dach* positive cells in *Platynereis* larval brain, marked by cluster 1 (red), cluster 2(blue) and cluster 3(green). The clustering was done with hopach package in R

statistical computing language. Sub clusters of cells are also visualized in dendrograms. (B) A horizontal line represents a cell and the proportional coloring represents the proportion reproducibility in cluster of that color. The graph is divided into three areas (marked by cluster 1, 2 and 3) for three distinct clusters. Bottom to Top shows Cluster 1(red), Cluster 2(blue) and Cluster 3(green). (C) visualizes the spatial distribution of cells belonging to each cluster in corresponding colors. White marks the average reference axonal scaffold of 48 hours old *Platynereis* larval brain. (D) shows the mediods of the three super clusters in multi-dimensional gene space. Row represents the coordinates of the mediods and column represents genes. Red square means a gene is present in the mediod of that particular cluster. The gene names end with a number, which represents the number of images of expression pattern that were used for generating the average expression pattern. For instance, Dach_6 means 6 different individual larvae were scanned for generating the average expression pattern for *Dach*.

2.1.2.3 Clustering of genes based on their spatial expression patterns

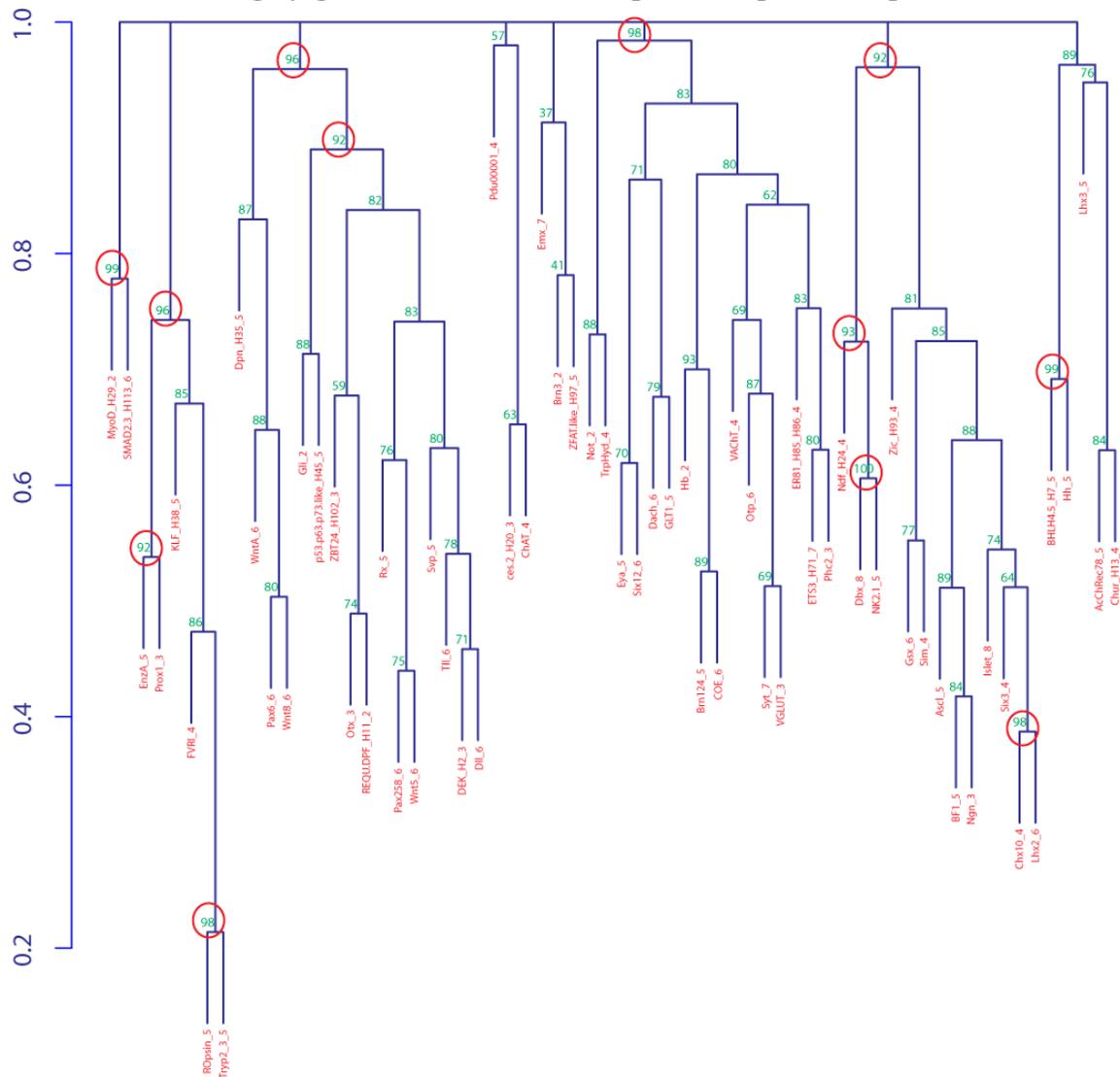


Figure 30. Hierarchical clustering of genes based on their spatial expression patterns.

The cellular expression profile data was used to investigate the relationship among genes based on their spatial expression patterns. The figure shows the dendrogram representation of the clusters. Also, the

statistical significance of clusters was calculated using multi-scale bootstrap analysis. Each node is given a statistical significance value on a scale of 100. Any value greater than 90 is marked by a red circle.

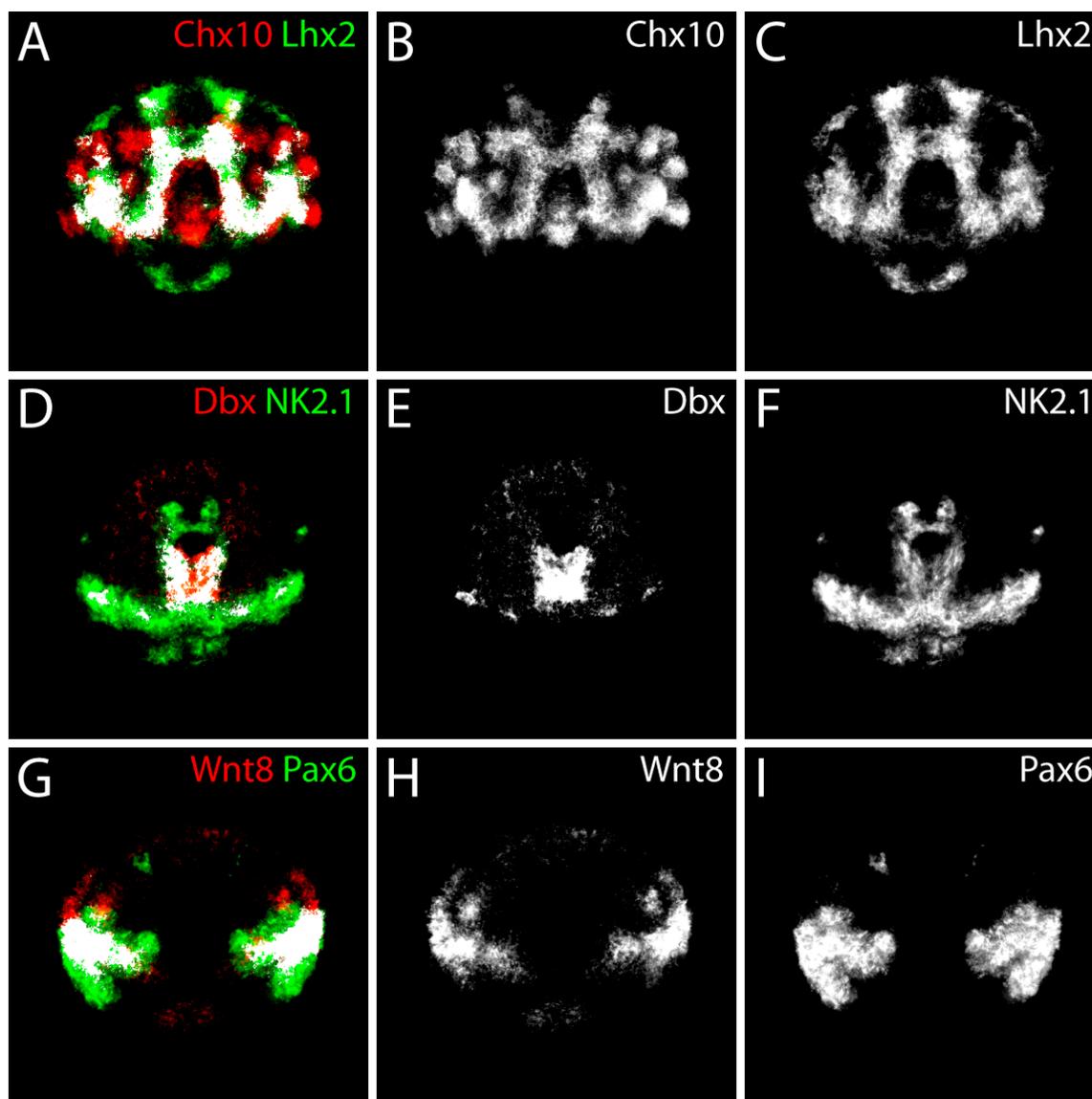


Figure 31. Case studies of clusters of genes

Examples of the spatial relationship of genes that cluster together with a significant statistical support, as shown in Figure 30. (A-C) shows the maximum z-projection of co-expression of *Chx10* (Red) and *Lhx2* (Green) in 48 hpf *Platynereis* larval brain. (D-F) shows the maximum z-projection of co-expression of *Dbx* (red) and *NK2.1* (green). (G-I) shows the maximum z-projection of co-expression of *Wnt8* (red) and *Pax6* (green). White pixels in A,D and G marks co-localization.

In addition to the investigation of relationships among *Platynereis* larval brain cells, I also studied the relationships of genes with each other, based on their spatial expression

pattern. This analysis should shed light on the possible co-regulated gene modules. Firstly, I did hierarchical clustering of genes using complete linkage and uncentered correlation (same as for the cells clustering) and then for the resulting clusters I calculated approximately unbiased p-values by multi-scale bootstrap resampling using pvclust (Suzuki and Shimodaira, 2006) to assess the support for the clusters. The results of the clustering analysis are shown in Figure 30. It is evident that there are many nodes in the clusters which are very well supported (marked with circles in Figure 30) by the data. For instance, some of the clusters with significant support are: the cluster of *EnzA*, *Prox1*, *KLF*, *FVRI*, *ROpsin* and *Tryp2_2* with a highly significant bootstrap p-value value of 96; the cluster of *Dpn*, *WntA*, *Pax6*, *Wnt8*, *Gli*, *p53-p63-p73-like*, *ZBT24*, *Otx*, *REQU-DPF*, *Rx*, *Pax258*, *Wnt5*, *Svp*, *Tll*, *Dek* and *Dll* with a highly significant bootstrap p-value of 96; the cluster of *Ndf*, *Dbx*, *NK2.1*, *Zic*, *Gsx*, *Sim*, *Ascl*, *BF1*, *Ngn*, *Islet*, *Six3*, *Chx10* and *Lhx2* with a bootstrap value of 92. The biological significance of the clusters is exemplified by the clustering of *EnzA*, *Prox1*, *KLF*, *FVRI*, *ROpsin* and *Tryp2_2*, where *EnzA*, *Prox1*, *FVRI*, *ROpsin* and *Tryp2_2* are known eye genes, with an additional gene *KLF* falling into the cluster. To further assess the quality of the clustering, I analyzed the spatial expression of some of the nodes with significant support from the data. For instance, the nodes of *Chx10* and *Lhx2*; *Dbx* and *NK2.1*; *Wnt8* and *Pax6* have bootstrap p-values of 98, 100 and 80 respectively. As shown in the Figure 31, it is evident that *Chx10* and *Lhx2* expressions largely overlap, *Dbx* expression is almost exclusively in *NK2.1* domain and *Wnt8* is expressed largely in *Pax6* domain.

The examples discussed in the above sections clearly corroborate the usefulness of WMISEP protocol for investigating the evolution of cell types and the defining genetic interactions. This is so far only a proof-of-principle that such analysis can be done and makes sense to some extent.

2.2 Evolution of Mushroom Body

2.2.1 *Platynereis dumerilii* Adult Brain anatomy

Polychaetes have been the subject of anatomical studies for over a century. Using the classical histological stainings, the neuroanatomists have accumulated an impressive wealth of information (Strausfeld et al., 1998). Though the classical studies provide us quite detailed neuroanatomy, they are often low-resolution. Besides, to properly understand the inter-relationships of various structural units in the brain, the three dimensional reconstructions are required. My first aim was to generate three-dimensional reconstructions of the adult *Platynereis* brain to better understand the anatomy and the connectivity among various structural units in the brain. I utilized the advanced fluorescence immunostaining techniques and confocal microscopy at high resolutions.

I started with the classical histological stainings (for example: Cason's trichrome) on cryo-sections of adult *Platynereis* to reproduce the classical knowledge. Subsequently I optimized an immunostaining protocol on the cryo-sections of *Platynereis* adult brain to visualize various neuropils present in the brain. Furthermore, I optimized a protocol for doing whole mount immunostainings and confocal microscopy of the adult *Platynereis* animals, which will allow me to look at the brain in its entirety. Figure 32A-H shows a series of images of 10 microns thick transverse physical sections (lateral to medial direction), acquired using the confocal microscopy. Blue color shows the signal from the nuclear stain dapi, which marks the cell bodies in the brain; the green color shows the immunostaining against acetylated tubulin, which marks the axonal scaffolds in the brain; and the red color contains the signal from staining of Phalloidin, which, among other things, marks the growth cones of the axon bundles. Figure 32I-L shows the horizontal optical sections of a confocal 3D image stack in dorsal to ventral orientation. In addition, I used the data shown in Figure 32I-L to do three-dimensional reconstructions of the adult *Platynereis* brain, which are shown in Figure 33. Several structures including, Mushroom Body, Pars lateralis, Pars intercerebralis, Eyes, Nuchal Organ, Gehrinloben and Antennal Nerve, could be identified in the adult *Platynereis* brain and are marked in Figure 33A.

Further, as shown in Figure 33D, I investigated the reconstructions from a ventro-lateral perspective to investigate the structure and connections of Palpae and Circum-esophageal connectives. It is evident that the circum-esophageal connectives further subdivides into R2, R3 commissures, which is in agreement with *Nereis* neuroanatomy descriptions (Orrhage, 1993).

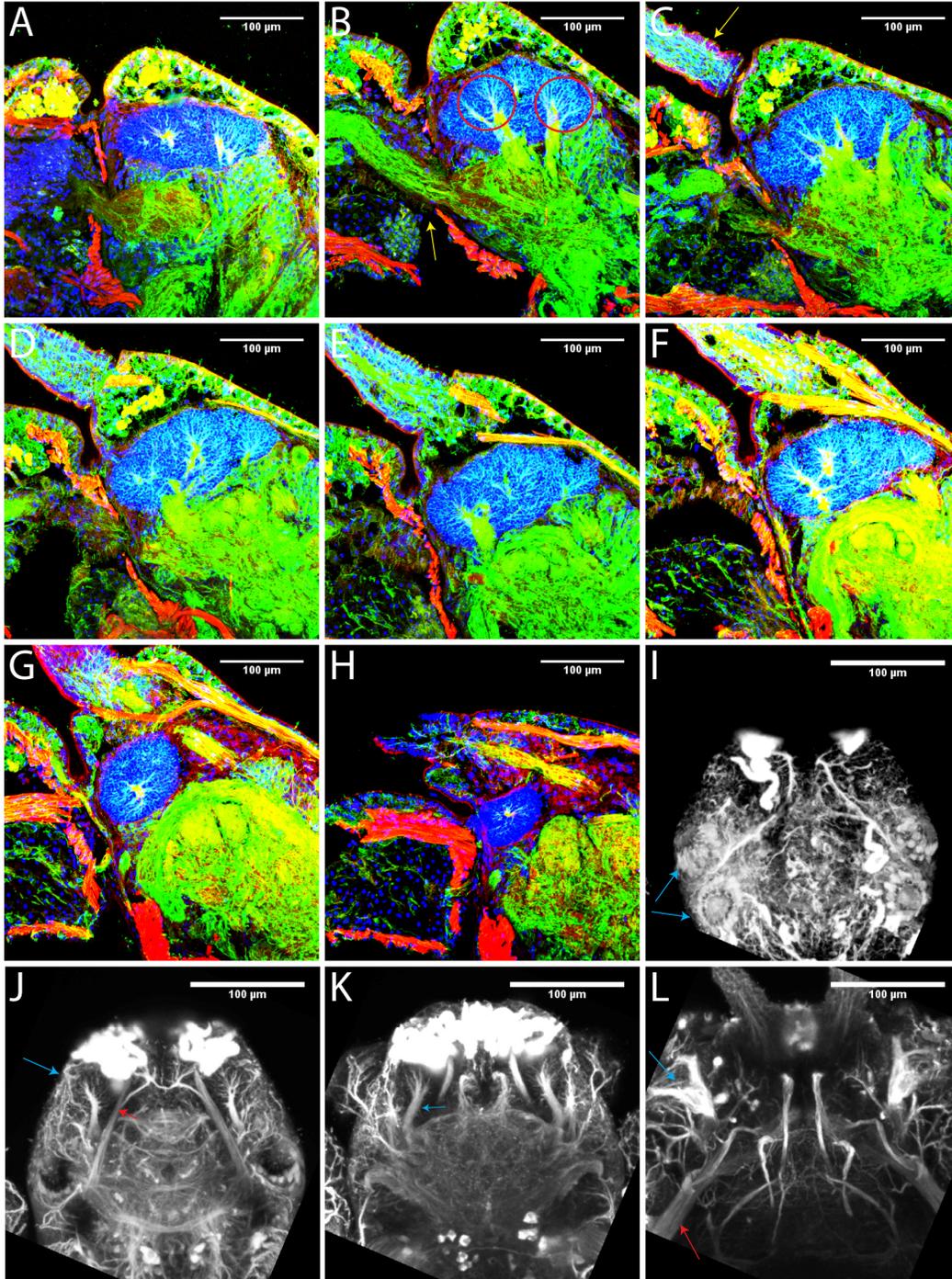


Figure 32. Anatomy of Adult *Platynereis* brain.

(A-H) shows a series of 10 microns thick physical sections from lateral to medial to lateral orientation. Anterior is top-left, ventral is bottom-left, dorsal is top-right and posterior is bottom-right. Blue is signal for nuclear stain dapi that marks all the cell bodies, green is immunostaining against acetylated tubulin that marks the axonal scaffold and red is staining using phalloidin which marks muscles and growth cones of neuropils. Red circles in B marks the two mushroom body pedunculi (dorsal and ventral). Yellow arrow in C points towards antenna. (I-L) shows a series (dorsal to ventral) of 35 microns thick optical sections of confocal image stack acquired from an adult *Platynereis* worm stained with an antibody against acetylated tubulin to visualize the axonal scaffold. Cyan arrows in I mark the eyes. Cyan and red arrows in J mark dorsal mushroom body and antennal nerve respectively. Cyan arrow in K marks the ventral mushroom body. Cyan and red arrows in L mark palpa and ventral nerve cord connectives respectively.

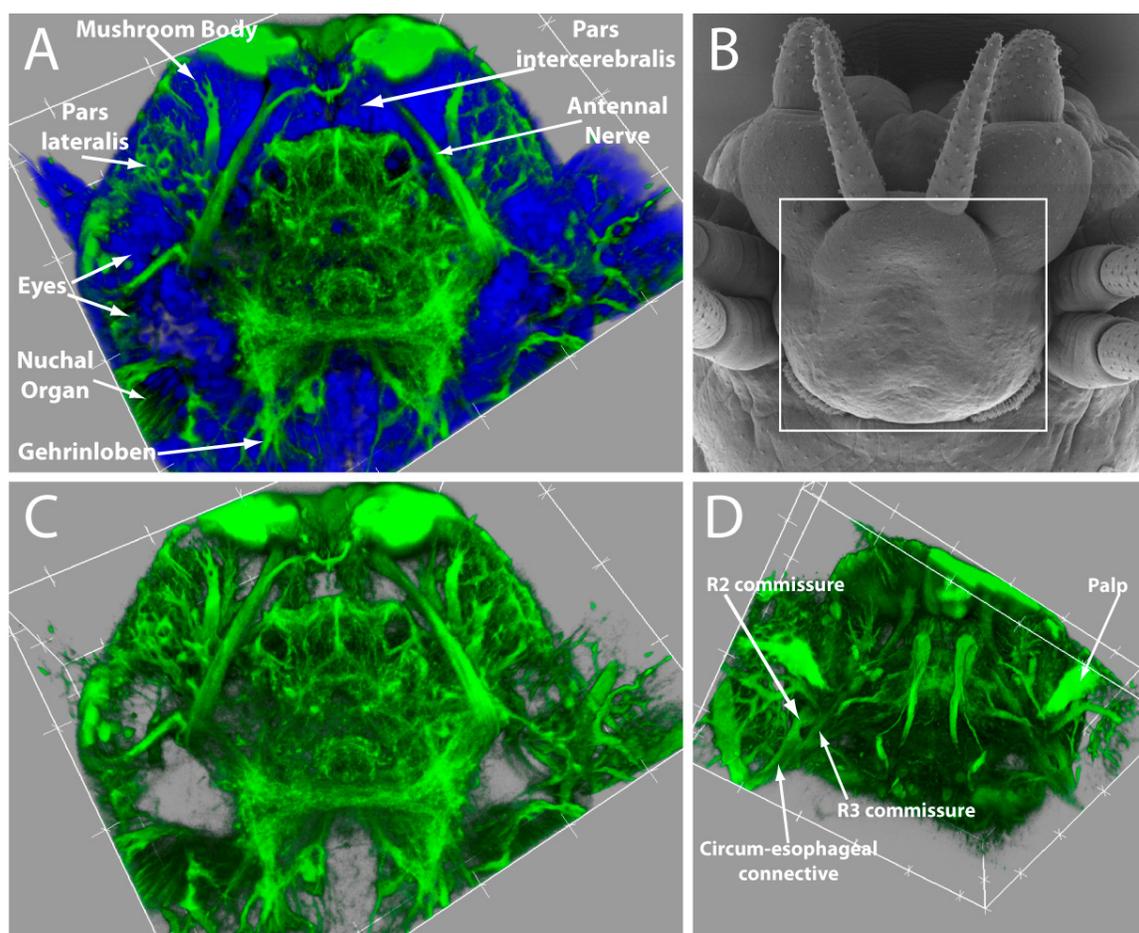


Figure 33. Three dimensional reconstruction of brain of adult *Platynereis dumerilii*.

(A,C) shows the three dimensional reconstructions of *Platynereis* adult brain. Green is immunostaining against acetylated tubulin and marks axonal scaffold. Blue is signal from nuclear staining dapi, marking the cell bodies. Major anatomical units that can be identified in *Platynereis* brain are: Mushroom Body, Pars lateralis, Pars intercerebralis, Eyes, Nuchal Organ, Gehrinloben and Antennal Nerve. (B) shows the Scanning Electron Microscopy picture (courtesy of Claus Nielson) of adult *Platynereis* head. The white square marks the brain in the head. (D) shows the three dimension reconstruction of adult *Platynereis* head from ventral-apical view. The anatomical units that could be identified are marked: Palp, Circum-esophageal connective, and R2, R3 commissures (in accordance with classical terminology).

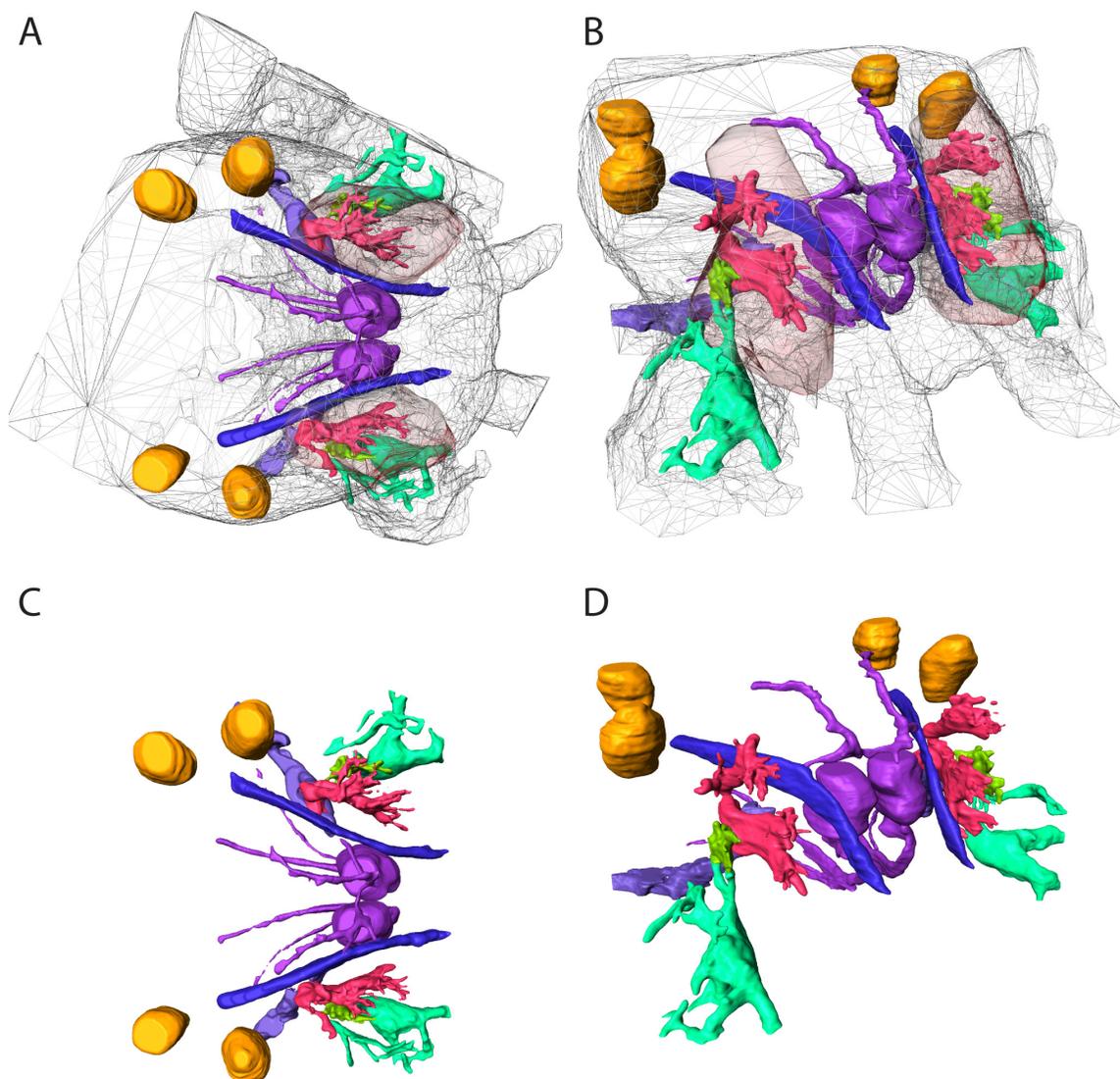


Figure 34. Three dimensional model of the brain of adult *Platynereis dumerilii*.

(A-D) shows three dimensional model of adult *Platynereis* brain in different orientations. (A): view from dorsal, (B): view from apical-lateral, (C,D): same views as A,B but without the body outline. The structures shown in detail are mushroom bodies pedunculi (red), mushroom bodies' cell bodies (transparent red), antennal nerve (dark blue), pars intercerebralis (purple), eyes (yellow), palpa (cyan) and ventral nerve cord connectives (light blue) and an unknown glomeruli-like structure (mediates the connection between palpa and mushroom bodies) in green. The model is based on the confocal images stack shown in Figure 32I-L. Amira (www.amiravis.com) was used for generating the model.

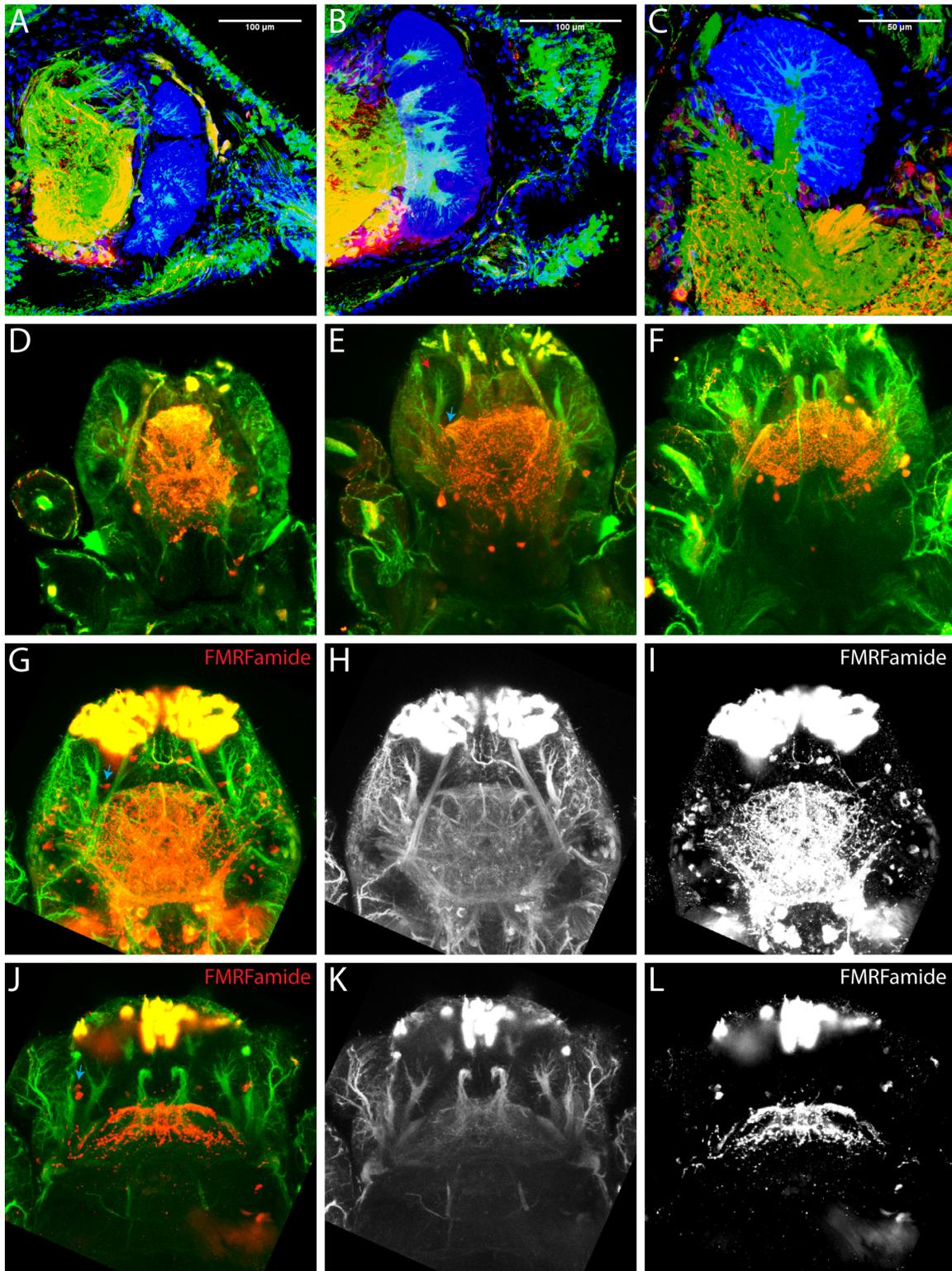


Figure 35. Serotonergic and FMRFamide positive neurons in relation to *Platynereis* mushroom bodies.

(A-B) shows two consecutive 40 microns thick physical sections. Anterior is to the left, dorsal is up and ventral is down. Blue represents nuclear stain Dapi, green represents immunostaining against acetylated

tubulin and red corresponds to immunostaining against serotonin. Scale bar represents 100 microns. (C) shows a high resolution view of mushroom body. Anterior is up, posterior is down and medial is to the right. Blue represents nuclear stain Dapi, green represents immunostaining against acetylated tubulin and red corresponds to immunostaining against serotonin. Scale bar represents 50 microns. It is apparent that mushroom bodies receive connections from serotonergic neurons. (D-F) shows series of horizontal optical sections generated by maximum z-projections of sub-stacks in dorsal to ventral orientation. Green represents immunostaining against acetylated tubulin and red corresponds to immunostaining against serotonin. The blue arrow in E points at the connection between serotonergic positive ventral nerve cord connectives and the medial lobe of mushroom body. (G-L) shows two consecutive optical sections showing spatial locations of FMRamide positive neurons in relation to mushroom bodies. Green represents immunostaining against acetylated tubulin and red corresponds to immunostaining against FMRamide. The blue arrows in G and J shows scattered FMRamide positive neurons near mushroom body pedunculi.

2.2.2 *Platynereis* Mushroom Body anatomy and Connections

The anterior part of the adult *Platynereis* brain contains a pair of large neuropils which anatomically resemble the insects' mushroom bodies and hence are also termed Mushroom Bodies (MBs) in *Platynereis*. To further investigate mushroom bodies and their connectivity with other parts of brain, I used confocal microscopy to acquire images of the adult *Platynereis* brain, which was immunostained with an antibody against acetylated tubulin to mark the axonal scaffolds and with nuclear stain dapi to mark the cell bodies. Subsequently, I did three dimensional modeling of the adult *Platynereis* brain using the confocal image stack as the template. As shown in Figure 34, the three dimensional model of *Platynereis* brain gives better understanding of mushroom body structure and it's relation to other structures in the brain. In Figure 34, the mushroom body pedunculus are represented in red and the cell bodies are shown in transparent Red. Additionally, the model also shows antennal nerve in dark blue, pars intercerebralis in purple, eyes in yellow, palpaes in cyan and ventral nerve cord in light blue. It is apparent that the mushroom bodies are located lateral to the antennal nerve, anterior to the eyes and dorsal to the palpaes. Further detailed morphological analysis showed that *Platynereis* mushroom bodies contain thousands of highly clustered globuli cells, which send axons posteriorily to form two axon bundles (Figure 34) (pedunculus) on each side of the brain. The pedunculi turn medially (Figure 34 and Figure 35C) at the posterior end to give rise to medial lobes. Interestingly, *Platynereis* mushroom bodies are proportionally very big, compared to the other structures in the brain. Since mushroom bodies in insects are implicated in higher order functions like learning and memory formation, it is tempting to

speculate about the role of mushroom bodies in controlling the complex behavior of *Platynereis dumerilii* (Evans, 1966a; Evans, 1966b; Evans, 1986). Furthermore, I investigated the connectivity of mushroom bodies with the other neuropils in the brain. As summarized in Figure 34, the detailed anatomical analysis demonstrated that the *Platynereis* mushroom bodies receive connections from palpa, which are putative chemosensory organs, via another neuropil structure, which resemble a glomurili (Figure 34). Additionally, *Platynereis* mushroom bodies receive connections from serotonergic neurons (Figure 35A-C), which make wiring like connections with the mushroom bodies pedunculi. Moreover, the immunostainings against neuropeptide FMRFamide (Figure 35G-L) shows the presence of a few scattered FMRFamide positive cells in the mushroom bodies. Further, I investigated the connections that would represent the output from mushroom bodies. As shown in Figure 35D-F, it is evident that the medial lobes of mushroom bodies make connections with ventral nerve cord connectives (marked by an arrow in Figure 35E), suggesting that the mushroom bodies might send signals to control the movement of the animal. In summary, *Platynereis* mushroom bodies behave as a sensory integration unit as it receives sensory input from sensory organs (for example Palpa) and drives the output via ventral nerve cord commissure.

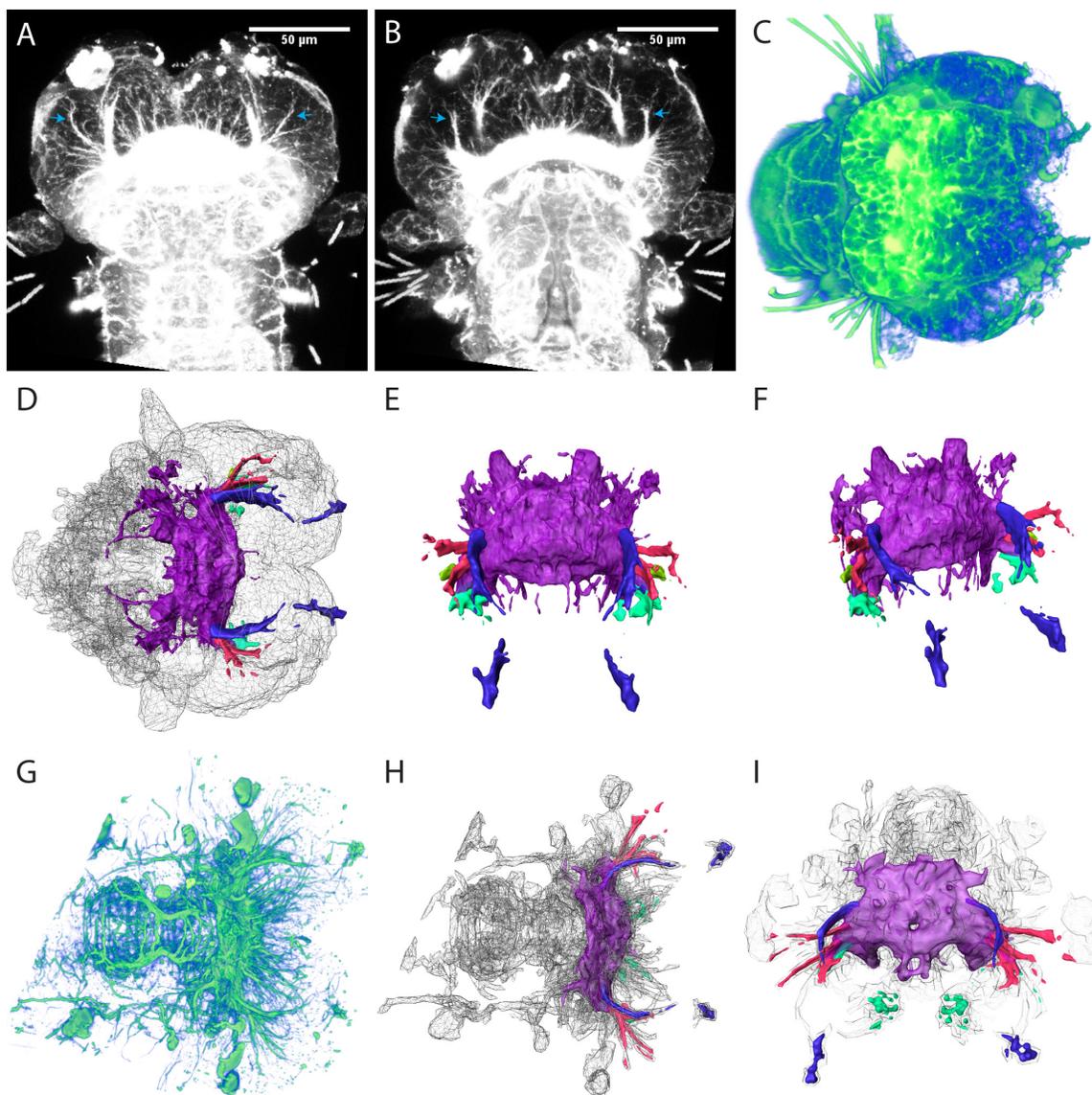


Figure 36. Tracing back the development of mushroom bodies anatomically.

(A,B) shows the maximum Z-projection of the confocal image stack, at the level of dorsal pedunculus and ventral pedunculus, of 10 days old young worm. The scale bar represents 50 microns. The dorsal and ventral mushroom body pedunculi are marked by cyan arrows in A and B. (C) three-dimensional reconstruction showing the outline of 10 days old young worm used for three dimensional modeling. (D-F) shows the three dimensional model of brain of 10 days old young worm. The pedunculi are colored red, the antennal nerve and the visible antenna outside the body is represented in dark blue. Cyan marks the putative pedunculi neuropils. (G-I) shows three-dimensional reconstruction and modeling of 3 days old larvae. The mushroom bodies neuropil are labeled in red and the antennal nerves are labeled in dark blue.

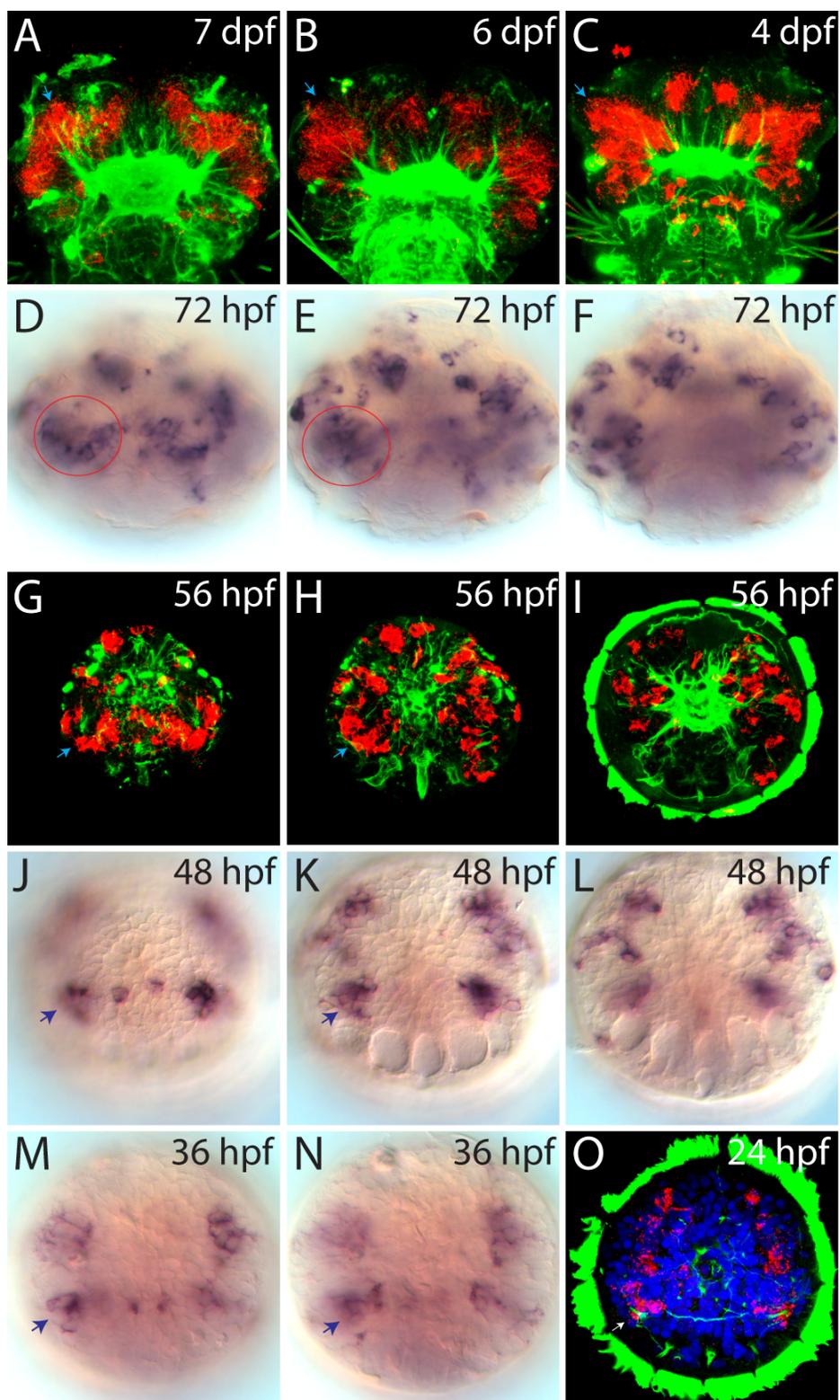


Figure 37. *Dach* is expressed in adult *Platynereis* mushroom bodies and their anlagen.

(A) *Dach* expression (red) in mushroom body in 7 days old young worm. Immunostaining against acetylated tubulin is shown in green. Blue arrow marks the mushroom body. (B) *Dach* expression (red) in

mushroom body in 6 days old young worm. Immunostaining against acetylated tubulin is shown in green. Blue arrow marks the mushroom body. (C) *Dach* expression (red) in mushroom body in 4 days old young worm. Immunostaining against acetylated tubulin is shown in green. Blue arrow marks the mushroom body. (D-F) shows the expression of *Dach* in 3 days old young worm in apical, D, to basal, F, direction. The red circle shows the mushroom body specific expression domain of *Dach*. (G-I) shows a series of optical sections to show the expression of *Dach* (red) in 56 hour old larvae. Green is immunostaining against acetylated tubulin. Blue arrow marks the expression of *Dach* in mushroom body anlagen. (J-L) shows a series of optical sections showing the expression of *Dach* in 48 hour old larvae. Blue arrow marks the expression of *Dach* in mushroom body anlagen. (M-N) shows a series of optical sections showing the expression of *Dach* in 36 hour old larvae. Blue arrow marks the expression of *Dach* in mushroom body anlagen. (O) shows the maximum Z-projection of expression of *Dach* (red) in 24 hour old embryo. Green is immunostaining against acetylated tubulin and blue is signal from nuclear stain dapi. White arrow marks the expression of *Dach* in mushroom body anlagen.

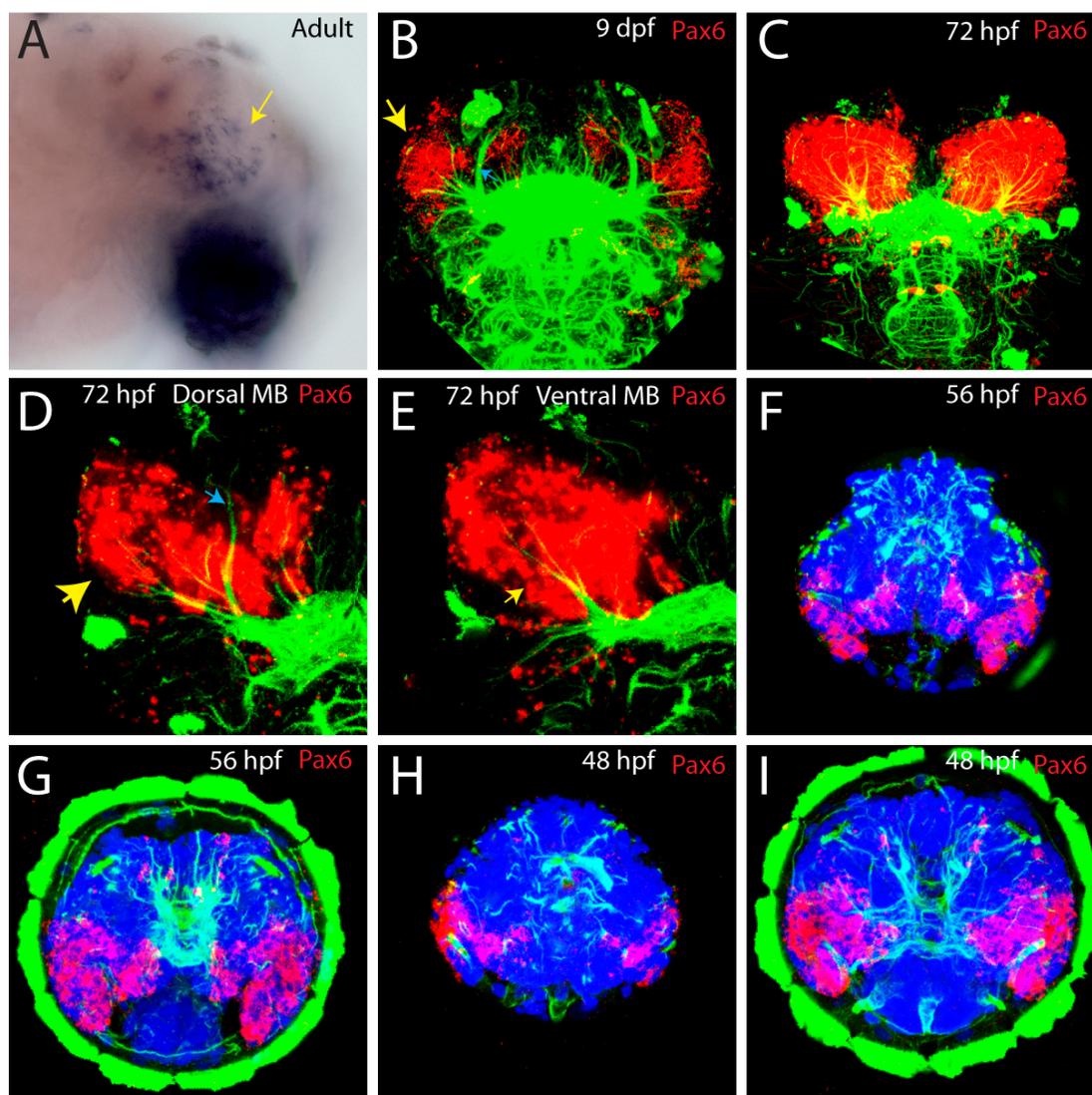


Figure 38. *Pax6* is expressed in adult *Platynereis* mushroom bodies and their anlagen.

(A) Lateral view of expression of *Pax6* in adult *Platynereis* mushroom body (yellow arrow) and palpa. Anterior is to the left, dorsal is up and ventral is down. (B) shows the expression of *Pax6* in mushroom

bodies (yellow arrow) of 9 days old young worm. Blue arrow marks the antennal nerve. (C-E) shows expression of *Pax6* in 3 days old larvae. C shows overall maximum z-projection, whereas D and E show the optical projections at the level of dorsal mushroom body and ventral mushroom body, respectively. Yellow arrows mark the mushroom body and blue arrow marks the antennal nerve. (F-G) shows expression of *Pax6* in 56 hour old larvae. Blue is signal from Dapi staining, green form acetyallate tubulin and red shows the expression pattern. Blue is signal from Dapi staining, green form acetyallate tubulin and red shows the expression pattern. It is apparent that the expression domain of *Pax6* covers mushroom body and palpa anlagen.

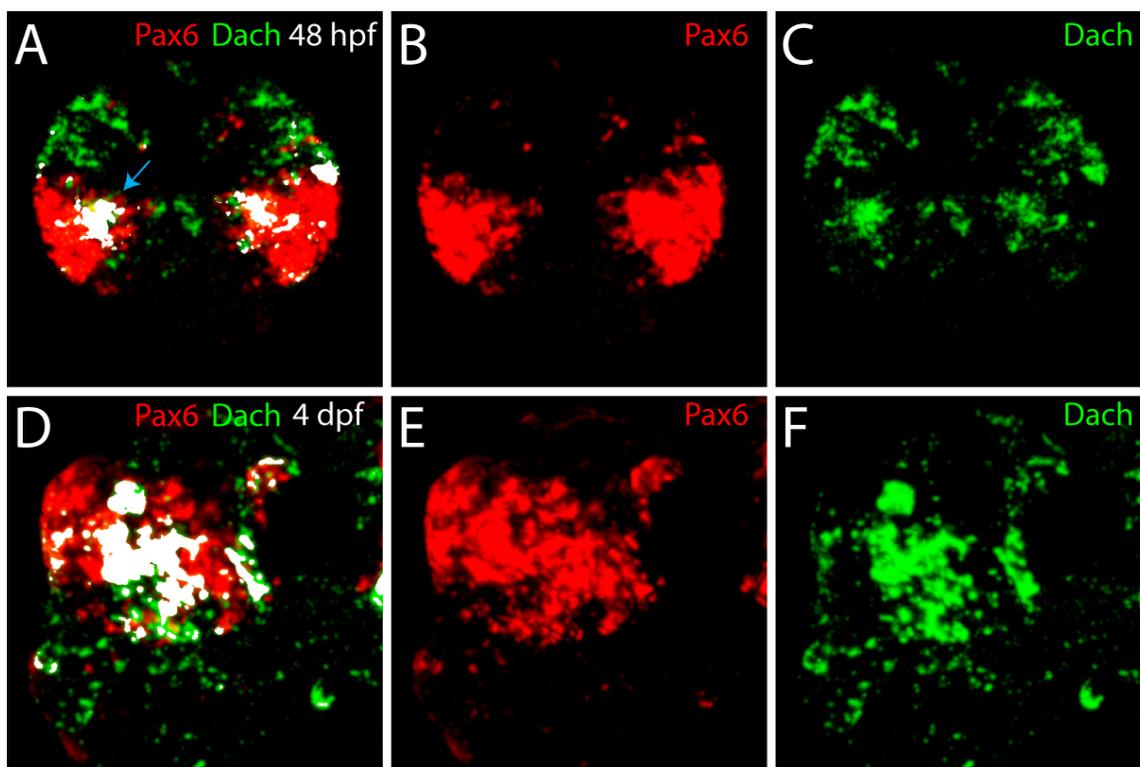


Figure 39. *Pax6* and *Dach* are co-expressed in *Platynereis* mushroom bodies and their anlagen

(A-C) Co-expression (arrow) of *Pax6* (Red) and *Dach* (green) in mushroom body anlagen of 48 hour old *Platynereis* larvae. Co-localized pixels are marked in white color. (D-F) shows the zoomed view of co-expression of *Dach* and *Pax6* in mushroom body cells of 4 days old *Platynereis* young worm. Co-localized pixels are marked in white color. Anterior is up, medial is to right and posterior is down.

2.2.3 *Platynereis* Mushroom Body Development

As shown in previous sections, *Platynereis dumerilii* has strikingly large mushroom bodies. The next question I asked was: How does *Platynereis* mushroom body develop and where are the anlagen located in the embryonic brain? To address this question I investigated the brain architecture of many intermediate larval stages of development, with the aim to trace back the origins of MB anatomically. I used immunostaining against

acetylated tubulin to mark the axonal scaffold and nuclear stain dapi to mark the cell bodies, and used confocal microscopy to acquire high resolution images for three dimensional reconstructions. The criteria I used to identify larval mushroom bodies was based on the fact that the *Platynereis* mushroom bodies are large, have characteristic pedunculi, and are located lateral to the antennal nerve, dorsal to palpa and anterior to the eyes. Using these criteria, I could easily identify the mushroom body neuropils in several larval stages, up to 3 days old larvae. The three dimensional reconstructions of 10 days old young worm and 3 days old larvae are presented in Figure 36, where mushroom body pedunculus are marked in red, antennal nerve and the visible antenna outside the body are marked in dark blue, the putative palpa nerve is marked in cyan, and the brain commissure is marked in purple. Subsequently, to identify the mushroom body anlagen in embryonic brain, I screened for marker genes that had expression in the larval mushroom body. Firstly, I used a candidate gene approach to investigate the expression patterns of several marker genes that are known to be expressed in mushroom bodies in *Drosophila*. As a result, I established that *Dach* (Figure 37) and *Pax6* (Figure 38), the earliest known marker genes for *Drosophila* mushroom body neuroblasts, are expressed in *Platynereis* adult and larval mushroom bodies. Moreover, I used double fluorescence WMISH (Whole Mount In Situ Hybridization) to test if *Dach* and *Pax6* are co-expressed in *Platynereis* mushroom bodies. Indeed, *Dach* and *Pax6* are co-expressed in mushroom bodies of 4 days old young worm (Figure 39D-F) and mushroom body anlagen in 48 hour old larvae (marked by cyan arrow in Figure 29A-B, marked by yellow circle in Figure 40A-B). Moreover, as reported by (Kurusu et al., 2000; Noveen et al., 2000), *Eya*(Figure 41R) and *So*(Figure 41S) are not expressed in mushroom body anlagen in 48 hour old *Platynereis* larval brain. After establishing the molecular markers and anatomical locations for mushroom bodies, I traced back the development of mushroom body to identify the mushroom body anlagen in *Platynereis* embryonic brain (Figure 37O).

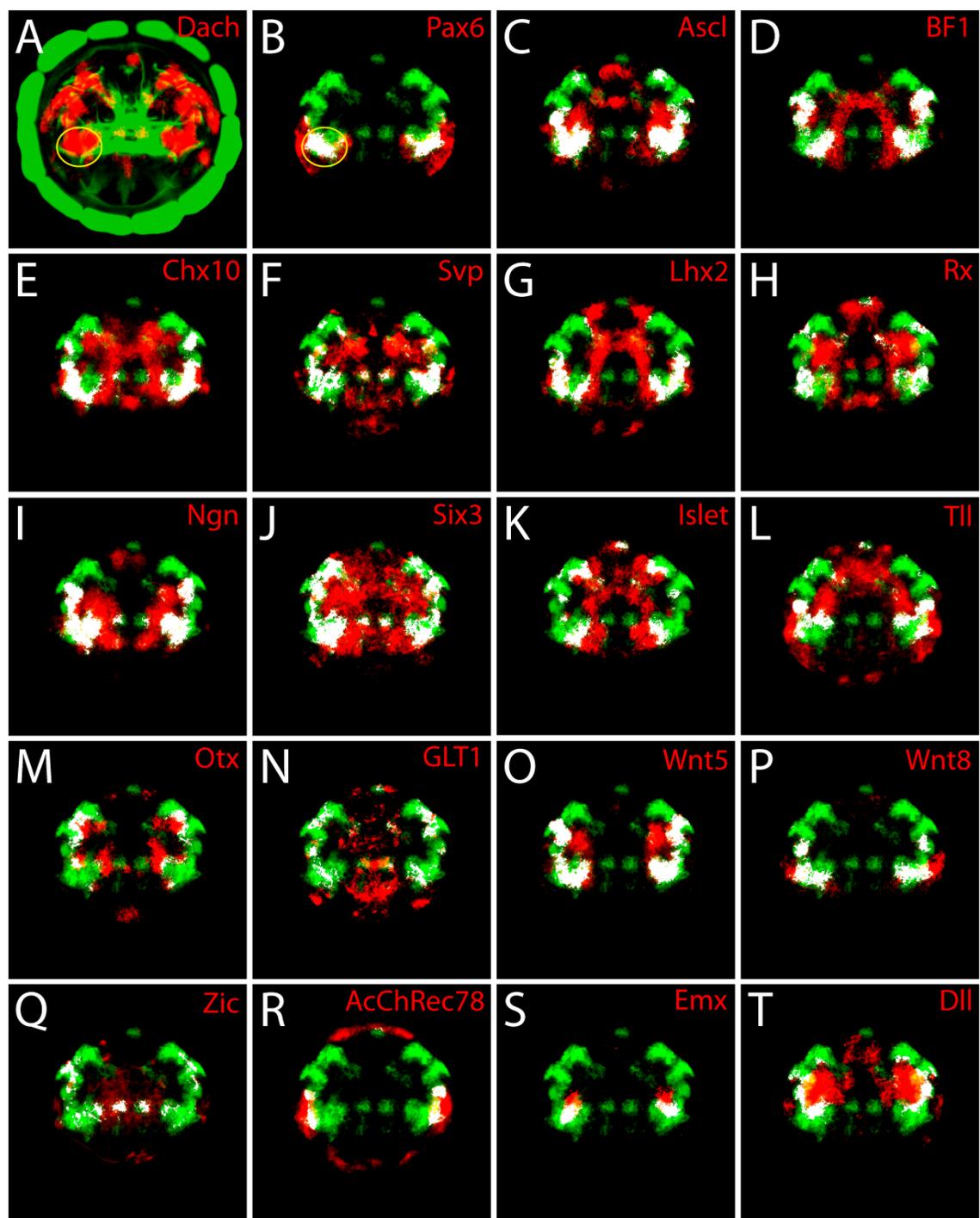


Figure 40. Screen for genes expressed in *Platynereis* mushroom body anlagen.

All the pictures are generated using WMISEP and are in apical view. (A) shows the maximum z-projection of the full expression domain of *Dach* (red) superimposed on average reference axonal scaffold (green) of 48 hour old *Platynereis* larval brain. The mushroom body anlagen are marked by yellow circle in A and B. (B-T) shows the co-expression *Dach* (green) and *Pax6*, *Ascl*, *BF1*, *Chx10*, *Svp*, *Lhx2*, *Rx*, *Ngn*, *Six3*, *Islet*, *Tll*, *Otx*, *GLT1*, *Wnt5*, *Wnt8*, *Zic*, *AcChRec78*, *Emx* and *Dll*. The co-localization is represented by white pixels. For sake of clarity, the images shown in B-T are maximum z-projections of confocal images sub-stack that includes *Dach* and *Pax6* co-expression domain only.

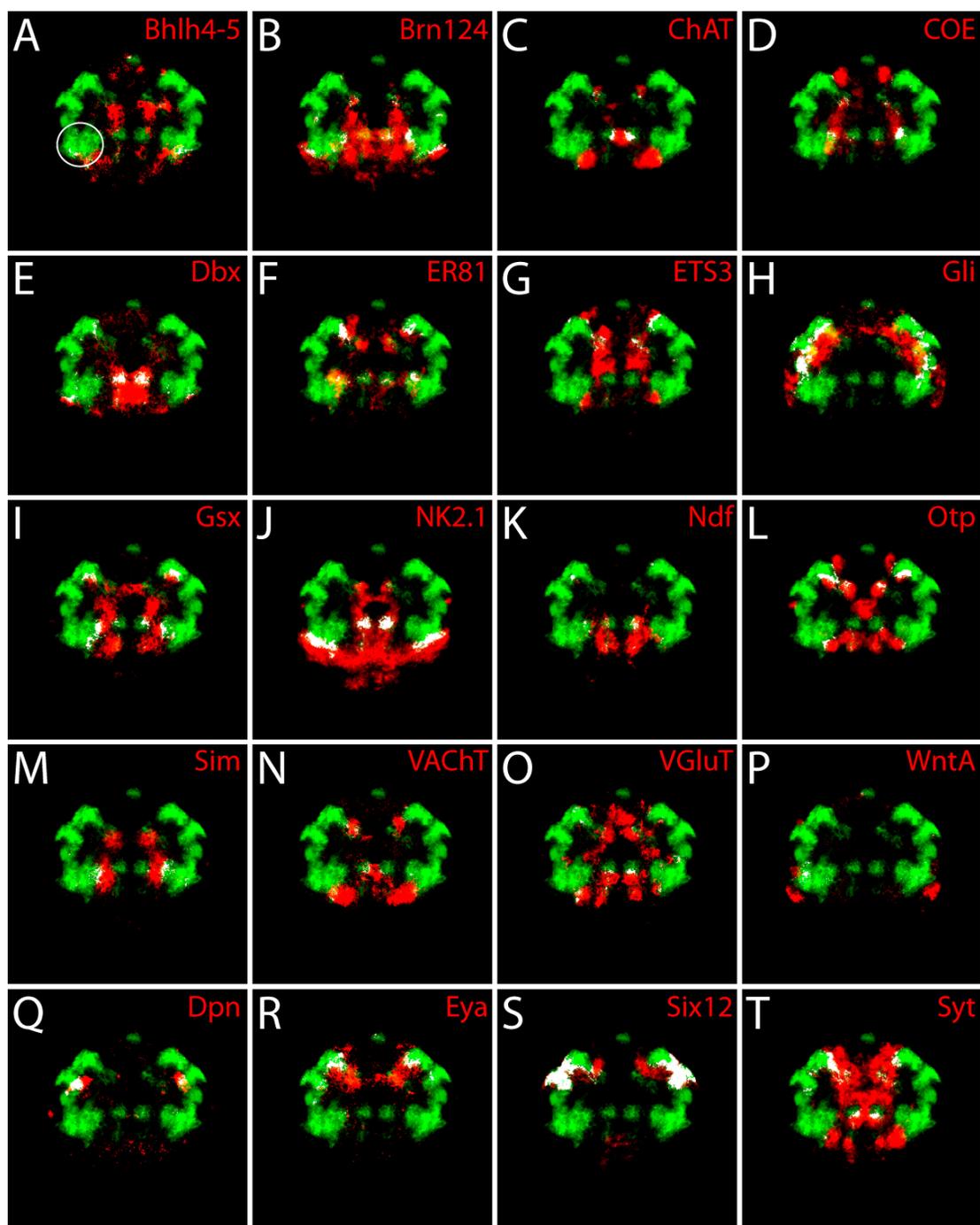


Figure 41. List for genes not (or marginally) expressed in *Platynereis* mushroom body anlagen.

All the pictures are generated using WMISEP and are in apical view. The figure shows the co-expression of several genes (as marked on individual pictures) with *Dach*. All pictures are in apical view. The mushroom body anlagen domain of *Dach* expression is marked by a white circle in A. It is apparent that none of these genes is significantly co-expressed with the mushroom body anlagen expression domain of *Dach*. For sake of clarity, all the images shown in A-T are maximum z-projections of confocal images sub-stack that includes *Dach* and *Pax6* co-expression domain only.

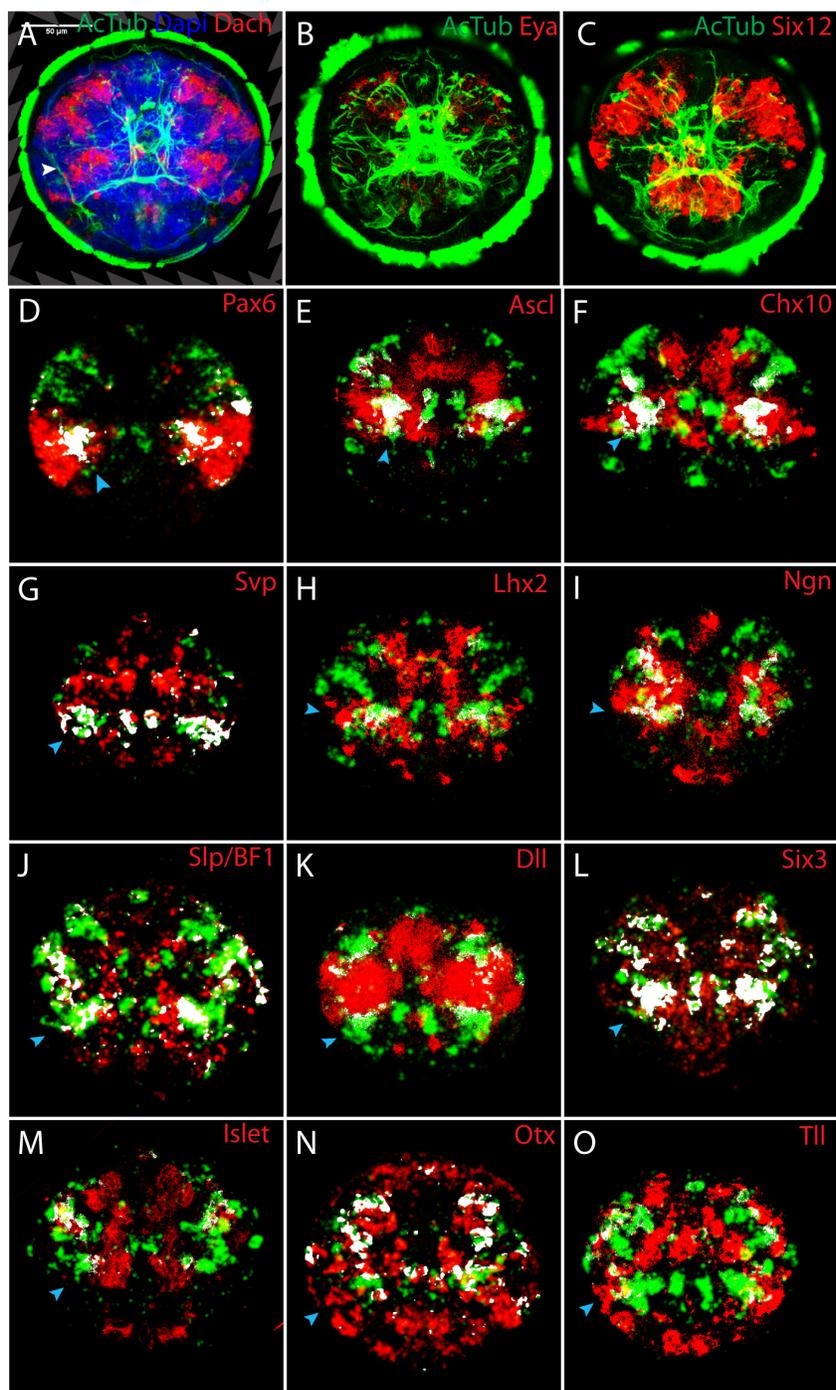


Figure 42. Verification of mushroom body molecular fingerprint genes

(A,B,C): Red shows the expression information acquired using reflection microscopy of NBT/BCIP precipitate, green shows the axonal scaffold visualized using immunostaining against acetylated tubulin and blue shows the dapi signal. (D,G,J,L,N): co-expression generated by using double fluorescence whole mount in situ hybridization (WMISH). (E,F,H,I,K,M,O): co-expression generated by using the combination of NBT/BCIP precipitate reflection microscopy (red) and fluorescence WMISH (green). The arrow heads (white and cyan) point towards the mushroom body anlagen cells. All images are maximum z-projections

of confocal image stacks and shows the expression in 48 hour old *Platynereis* larval brain in apical view (dorsal is up and ventral is down). *Eya*(B) and *six12/so*(C) are not expressed in mushroom body anlagen whereas *Pax6*(D), *Ascl*(E), *Chx10*(F), *Svp*(G), *Lhx2*(H), *Ngn*(I), *slp/BF1*(J), *Dll*(K), *Six3*(L), *Islet*(M), *Otx*(N) and *Tll*(O) are expressed in all or subsets of mushroom body anlagen cells. This figure also illustrates the limitations of using double fluorescence wimsh (noisy signal), and of combined reflection and fluorescence microscopy (quenching of fluorescence signal, for example *Dll* in K where part of *Dach* expression domains are shaded) for co-expression studies.

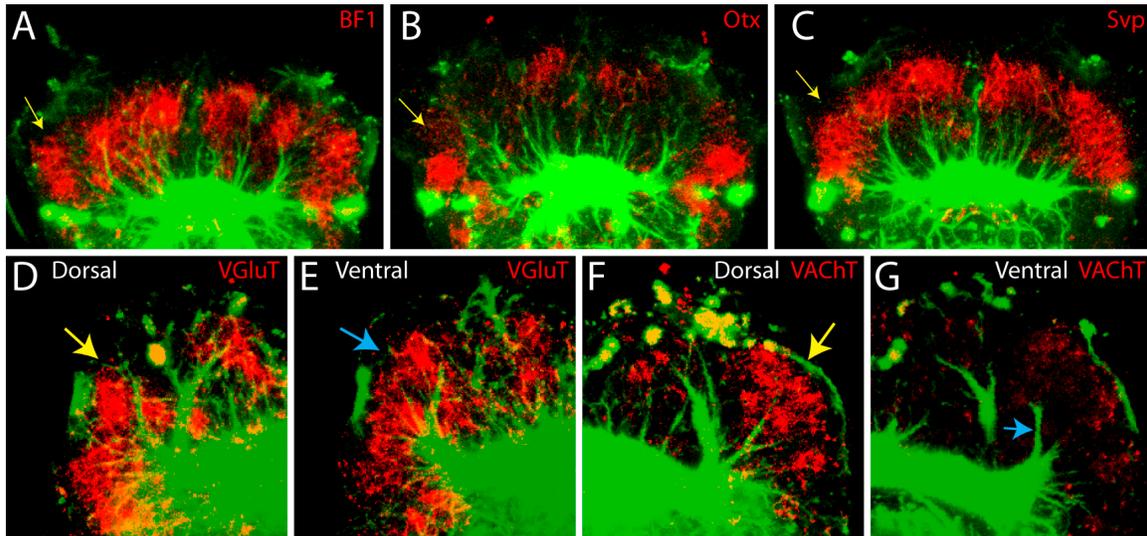


Figure 43. Expression of BF1, Otx, Svp, VGluT and VAcHT in mushroom body in older stages.

(A-C) shows dorsal views, anterior is up and posterior down. (A) Expression of BF1 in 4 days old *Platynereis* young worm. Yellow arrow marks the expression in mushroom body. (B) Expression of Otx in 4 days old *Platynereis* young worm. Yellow arrow marks the expression in mushroom body. (C) Expression of *Svp* in 4 days old *Platynereis* young worm. Yellow arrow marks the expression in mushroom body. (D, E) Expression of *VGluT* (marker for glutamatergic neurons) in 10 days old *Platynereis* young worm at the level of dorsal and ventral mushroom body. Yellow and cyan arrows mark the expression in dorsal and ventral mushroom body. Anterior is up, medial is to right and posterior is down. (F, G) Expression of *VAcHT* (marker for cholinergic neurons) in 10 days old *Platynereis* young worm at the level of dorsal and ventral mushroom body. Yellow and cyan arrows mark the expression in dorsal and ventral mushroom body. Anterior is up, medial is to left and posterior is down.

2.2.4 *Platynereis* Mushroom Body Molecular Fingerprint

Having identified the Mushroom Body anlagen in *Platynereis* embryonic brain, the next question I asked was: What is the molecular fingerprint of *Platynereis* Mushroom Body?

Firstly, I used the newly developed WMISEP protocol to screen for genes that are expressed in mushroom body anlagen in the 48 hour old larval brain, marked by a specific expression domain of *Dach* (Figure 40A, yellow circle) which co-expresses *Pax6*(Figure 40B, yellow circle) but not *Eya*(Figure 41R) and *Six12/so*(Figure S). As

shown in Figure 40, a number of genes are co-expressed with mushroom body *Dach* expression domain in 48 hour old *Platynereis* larval brain, including *Ascl*, *BF1*, *Chx10*, *Svp*, *Lhx2*, *Rx*, *Ngn*, *Six3*, *Islet*, *Tll*, *Otx*(in some cells), *GLT1*, *Wnt5*, *Wnt8*, *Zic*, *AcChRec78*(few cells), *Emx*(few cells) and *Dll*(making overlapping boundary). Moreover, I could identify several other genes as well, which are not expressed in mushroom body anlagen (Figure 41), and hence expanding the definition of molecular fingerprint to include the list of genes which are not expressed. Subsequently, I validated many of these genes by using the two-color whole mount in situ hybridization procedure. As shown in Figure 42, it is evident that the co-expression generated using double fluorescence whole mount in situ hybridization procedure matches very well (see legend Figure 42) with the co-expression data generated with WMISEP, further validating the WMISEP protocol. Moreover, I also investigated some of the transcription factors and the differentiation marker genes in older larval stages. As shown in Figure 43A-C, I investigated the expression of *BF1*, *Otx* and *Svp* in 4 days old young worms. Similar to their expression in 48 hpf larvae, these genes are indeed expressed in the mushroom bodies cells, among other parts of the brain. Further, I tested some differentiation marker genes, including *VGluT* (a marker for glutamatergic neurons) and *VACHT* (a marker for Cholinergic neurons), in 10 days old young worms. By analyzing the expression in dorsal(Figure 43D,F) and ventral mushroom bodies(Figure 43E,G), it is evident that the majority of mushroom body neurons are glutamatergic, with some cholinergic neurons in the dorsal mushroom body. Taking all together, the unique molecular fingerprint (minimal) of *Platynereis* mushroom body can be summarized as: (i) Transcription Factors: *Ascl*, *BF1*, *Dach*, *Chx10*, *Islet*, *Lhx2*, *Ngn*, *Pax6*, *Rx*, *Six3*, *Svp*, *Tll*, *Zic*, *Otx* (few cells), *Emx* (few cells) (ii) Signaling molecules: *Wnt5*, *Wnt8* (iii) Neurotransmitter receptors and transporters: *AcChRec78*(few cells), *VGluT*(differentiated MB), *VACHT*(in dorsal differentiated MB), Serotonin Receptors (most likely as MB receives connections from Serotonergic neurons, Figure 35C). Moreover, a diploma student in our laboratory (Benjamin Backfisch), whom I was supervising, used a functional morpholino against *Pax6* to interfere with its function (Diploma Thesis, Benjamin Backfisch, University of Heidelberg). The preliminary results indicate that *Pax6* does not regulate *Dach*

expression in *Platynereis* mushroom body cells, similar to *Drosophila* mushroom bodies development (Kurusu et al., 2000; Noveen et al., 2000).

2.2.5 Common origin of Insects and Platynereis Mushroom Body

The detailed investigation of mushroom body anatomy, development and molecular fingerprint presents the opportunity to compare them to insect mushroom bodies. As discussed in the introduction of this thesis, most of the details about insect mushroom bodies come from study in *Drosophila*, Honey Bee, Cricket and Cockroaches. The basic characteristic of insect mushroom bodies is the presence of highly clustered small-diameter globuli cells that send parallel axons to make stalk like pdenculi which often end in medial lobes. Additionally, some species also possess additional structures like calyces and additional lobes. As discussed above, *Platynereis* mushroom bodies contain thousands of tightly clustered globuli-like cells, which send axons posteriorly to form two axon bundles on each side of the brain, and turn medially at the posterior end to give rise to medial lobes. Therefore, the anatomical comparison corroborates the similarities of insects and *Platynereis* mushroom bodies, as has been suggested before (Muller, 1973; Strausfeld et al., 1998). Furthermore, several studies in *Drosophila* have identified many genes involved in the proper development of mushroom bodies. Firstly, (Kurusu et al., 2000; Noveen et al., 2000; Urbach and Technau, 2003) have reported that mushroom body neuroblasts in *Drosophila* are uniquely defined by the combination of *Dach*, *Pax6*, *BFI* and *Svp*. Additionally, *Eya* and *So* are not expressed in mushroom body neuroblasts (Kurusu et al., 2000; Noveen et al., 2000), and since *Eya* and *So* are part of the eye specification network (alongwith *Dach* and *Pax6*), the absence of expression of *Eya* and *so* makes the mushroom body neuroblasts different from the eye precursors. Moreover, functional interference experiments with *Dach* and *Pax6* demonstrate that their expression is absolutely required (Kurusu et al., 2000; Noveen et al., 2000) for the proper development of mushroom bodies, and that *Dach* and *Pax6* do not regulate each other's expression in mushroom body neuroblasts, contrary to their roles in eye development. The first question I asked: Do *Platynereis* mushroom body anlagen express the same

unique combination of genes? Indeed, as shown in Figure 40, the mushroom body anlagen in *Platynereis* larval brain expresses the combination of *Dach* (yellow circle in Figure 40A), *Pax6*(Figure 40B, Figure 42D), *BF1* (Figure 40D, Figure 42J) and *Svp* (Figure 40F, Figure 42G). In addition *Eya* (Figure 41R, Figure 42B) and *So* (Figure 41S, Figure 42C) are not expressed in mushroom body anlagen, as in *Drosophila*. Furthermore, I investigated many additional genes reported in literature to be expressed in mushroom bodies. The detailed comparison of insect and *Platynereis* mushroom bodies molecular fingerprints is as follows:

Insect's mushroom bodies molecular fingerprint (taken from the Introduction section, genes for which both *Platynereis* and Insects have information are underlined):

- I. Transcription factors: *Ascl*, *BF1*, *Brf*, *Dach*, *fkh*, *Hr46*, *jing*, *Lhx2*, *Otx*, *Pax6*, *Rx*, *Svp*, *tll*
- II. Neurotransmitters: glutamate, aspartate, taurine, neuropeptide F, GCCK (Honey Bee, Cockroach), Nitric Oxide (NAPDH histochemistry analysis).
Neurotransmitter Receptors: Serotonin Receptors, Acetyl Choline Receptors, GABA receptors, Dopamine receptors, Octopamine receptor
- III. Signaling Molecules: *Wnt5*

Platynereis mushroom bodies molecular fingerprint (genes for which both *Platynereis* and Insects have information are underlined):

- I. Transcription Factors: *Ascl*, *BF1*, *Dach*, *Chx10*, *Islet*, *Lhx2*, *Ngn*, *Otx*, *Pax6*, *Rx*, *Six3*, *Svp*, *Tll*, *Zic*, *Emx* (few cells), *Otx*(few cells)
- II. Neurotransmitter receptors and transporters: *AcChRec78* (*Acetyl Choline receptor 7/8*), *VGluT*(*Glutamatergic neurons*), *VAcHT*, Serotonin Receptors (indirect most likely inference as MB receives connections from Serotonergic neurons).
- III. Signaling molecules: *Wnt5*, *Wnt8*

Though for some of the genes, the information is not available in either insect or *Platynereis* side, but already the overall molecular fingerprint look very similar. The transcription factors that are common in insect and *Platynereis* mushroom body molecular fingerprints are: *Ascl*, *BF1*, *Dach*, *Chx10*, *Svp*, *Lhx2*, *Rx*, *Ngn*, *Six3*, *Islet*, *Tll*,

and *Otx*. Moreover, both insects (Grillenzoni et al., 2007) and *Platynereis* mushroom bodies express *Wnt5* (Figure 40O). In addition, insect mushroom bodies are known to use glutamate as neurotransmitter (Farris, 2005) and indeed *Platynereis* mushroom bodies also uses glutamate as neurotransmitter as indicated by the expression of *VGluT* (Vesicular Glutamate Transporter) (Figure 43D,E), a marker for glutamatergic neurons. There are a couple of mismatches as well. Firstly, *Otx/Otd* is reported to be expressed in all the four neuroblasts which give rise to *Drosophila* mushroom body (Urbach and Technau, 2003), whereas in *Platynereis* I found *Otx* to be expressed in only a fraction of mushroom body anlagen cells (Figure 40M). Second, *Emx/Ems* is not expressed in any of the four mushroom body neuroblasts in *Drosophila* (Urbach and Technau, 2003), whereas a small fraction of mushroom body anlagen cells appear to be expressing *Emx*. Together, the overall molecular fingerprint and anatomy similarity shows the deep homology of insect and *Platynereis* mushroom bodies, suggesting that the last common ancestor of Ecdysozoans (e.g. *Drosophila*) and Lophotrochozoans (e.g. *Platynereis*) already possessed mushroom body like cell types.

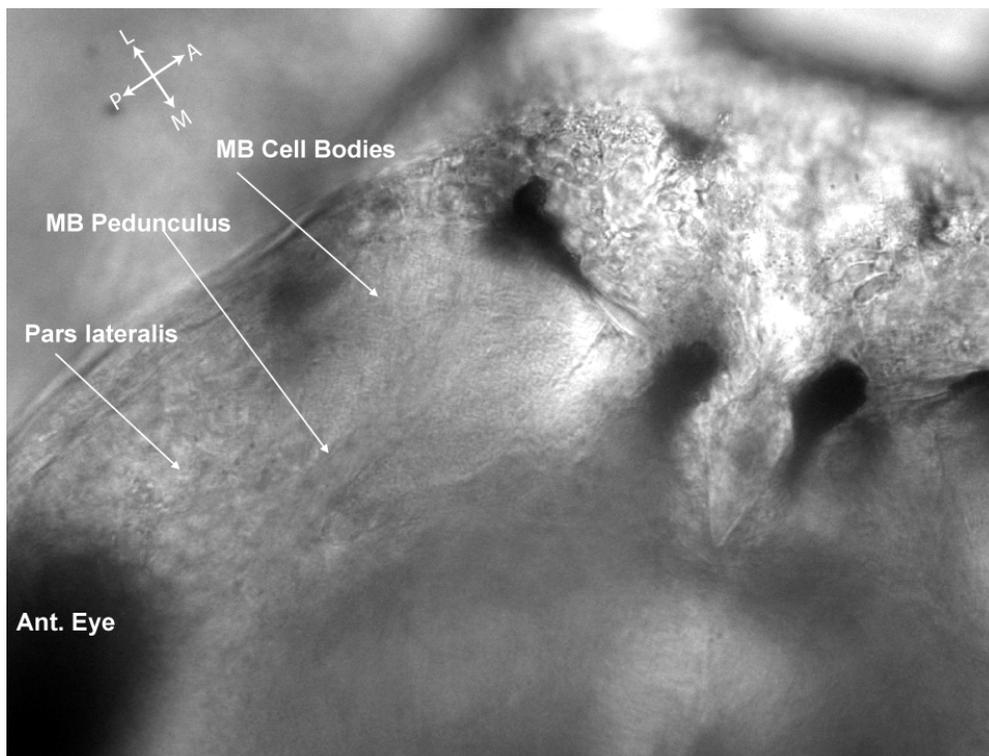


Figure 44. Visualizing mushroom bodies in alive *Platynereis* worms

This figure shows DIC white light picture of the head of alive adult *Platynereis* worm. The structures in the brain that can be recognized are marked.

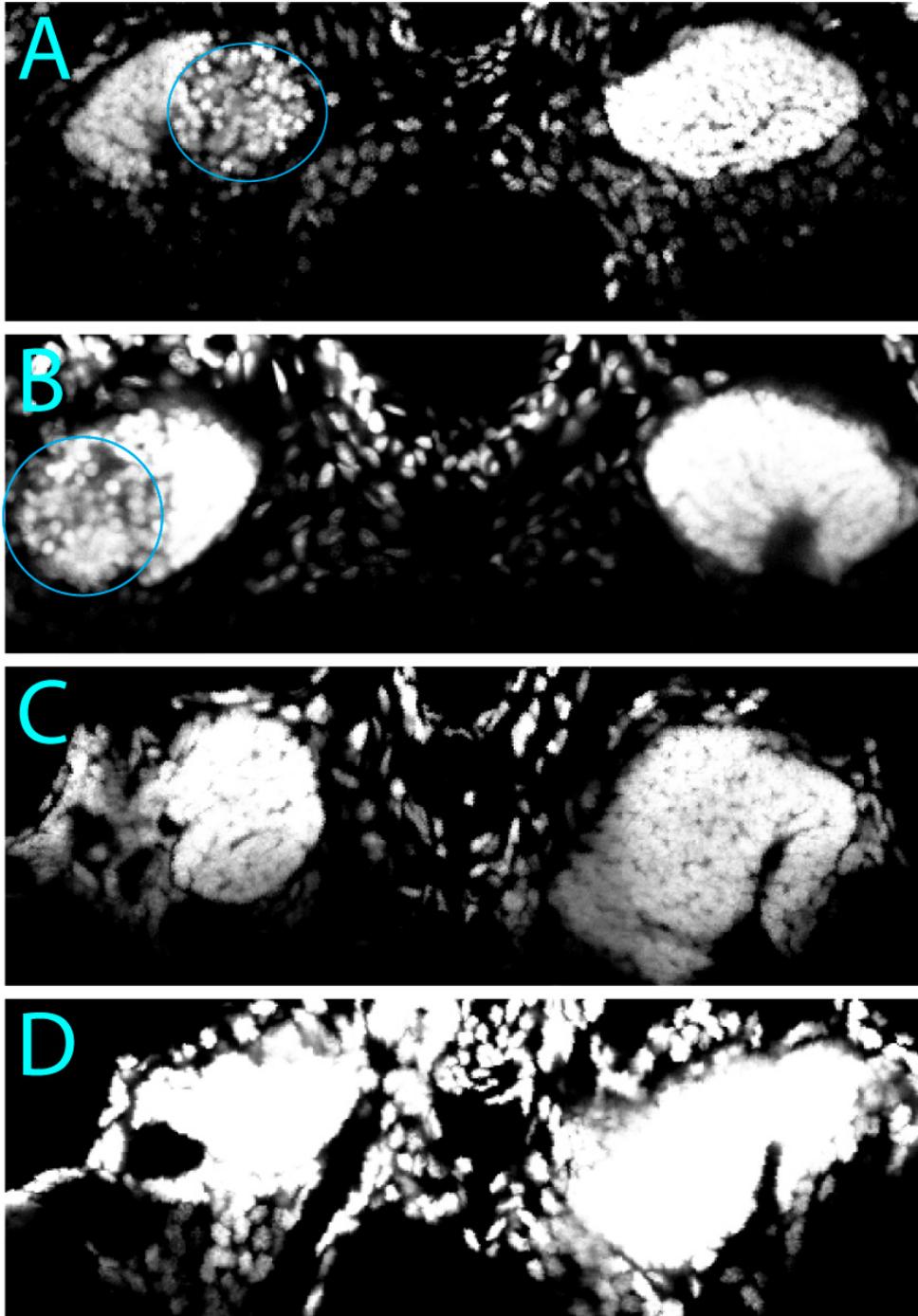


Figure 45. Ablation of *Platynereis* mushroom bodies using LASER cutter.

(A,B) shows two examples of ablation of mushroom bodies. Only left side mushroom bodies were ablated and right were left as control. The cyan circle marks the damaged cells. The pictures are in dorsal view, with anterior up and posterior down. (C,D) shows the recovery of ablated mushroom bodies one week after the ablation experiment. Left side mushroom bodies were ablated, and the worms were fixed after 1 week of recovery. Though, there are no dead cells visible, the size remains significantly smaller than the control (right side).

2.2.6 Towards deciphering the function of *Platynereis* mushroom body

Platynereis dumerilii has relatively large mushroom bodies. What is the function of these large mushroom bodies? To address this question, the first step was to establish a method to interfere with the function of *Platynereis* mushroom bodies. I tested many approaches including chemical ablations (de Belle and Heisenberg, 1994) and LASER surgery. The chemical ablation method has been successfully used to ablate mushroom bodies in *Drosophila*, utilizing the fact that there is a time window in *Drosophila* embryonic development when only mushroom body neuroblasts are proliferating; and using hydroxyurea to inhibit the proliferation in this time window resulted in severely compromised mushroom bodies. The first step was to investigate the existence of any such time window in *Platynereis* embryonic development when only mushroom body anlage is proliferating. I used BrdU incubation assays at several time points of *Platynereis* embryonic development, but could not identify any time window where only mushroom body anlagen were proliferating. Alternatively, one of the many advantages of *Platynereis* as a model system is that they are semi-transparent, and hence amenable to LASER based manipulations. Therefore, firstly I tested if adult *Platynereis* mushroom bodies can be seen under white light microscopes. And indeed, as shown in Figure 44, it is possible to see the mushroom bodies in living adult animals, which prompted me to test a LASER surgery system established by Julien Colombelli (Stelzer Group, EMBL Heidelberg). Together with Julien Colombelli, we used the LASER cutter based surgery approach to ablate the mushroom body cells. Subsequently, I used nuclear stain dapi to visualize the mushroom bodies and as illustrated in Figure 45, the mushroom body cells were specifically ablated without any significant side effect. We ablated the left side mushroom body cells and treated the right mushroom body as control. Figure 45A,B shows the ablated mushroom bodies, visualized approximately one hour after the experiment and Figure 45C,D shows the ablated mushroom bodies visualized after one week of recovery. There are no remaining damaged cells visible after one week of recovery, indicating the role of the immune system of *Platynereis* brain to clear the dead neurons. Also, the size of the recovered mushroom (probably as a result of regeneration)

bodies is significantly smaller than the control, suggesting that the LASER ablation approach can indeed be used for studying the mushroom body function (at least for a couple of days long assays) in *Platynereis*. The next step would be to establish a set of comprehensive behavioral assays to identify the role of mushroom bodies in *Platynereis* behavior (Evans, 1966a; Evans, 1966b; Evans, 1986).

2.3 Evolution of vertebrate telencephalon

2.3.1 Vertebrate telencephalon regionalization genes are expressed in similar topological order in *Platynereis dumerilii*

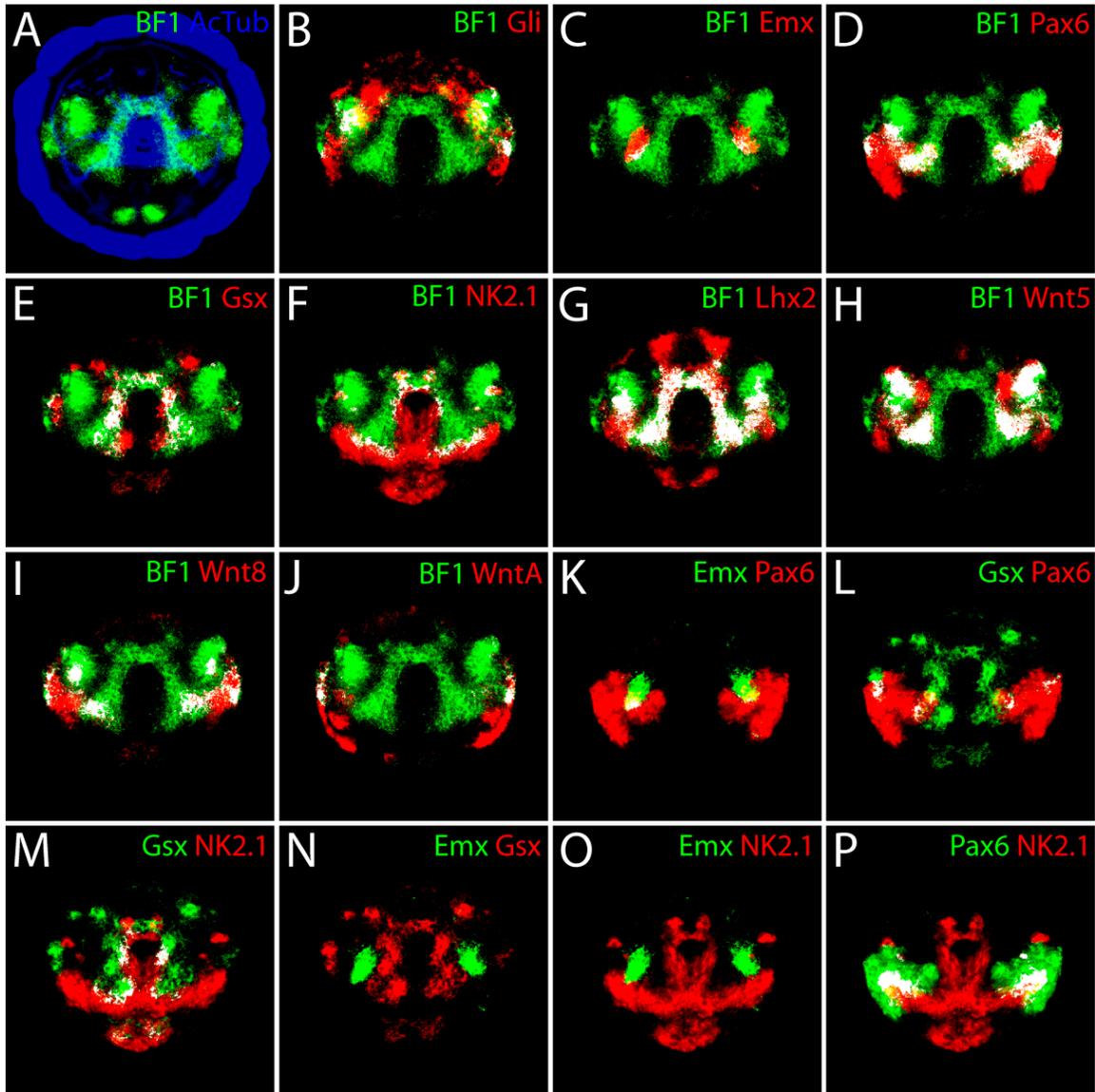


Figure 46. Expression of vertebrate telencephalon regionalization genes in *Platynereis* 48 hour larvae.

All the pictures are generated using WMISEP and are in apical view. White color marks the colocalized pixels. (A) *BF1* expression (green) superimposed over the average reference axonal scaffold (blue) of 48 hour old *Platynereis* larval brain. (B-J) shows the expression of *Gli*, *Emx*, *Pax6*, *Gsx*, *NK2.1*, *Lhx2*, *Wnt5*, *Wnt8* and *WntA* in comparison with *BF1*. (K) spatial co-expression of *Emx* (green) and *Pax6* (red). (L) co-expression of *Gsx* (green) and *Pax6* (red). (M) co-expression of *Gsx* (green) and *NK2.1* (red). (N) co-expression of *Emx* (green) and *Gsx* (red). (O) co-expression of *Emx*(green) and *NK2.1* (red). (P) co-expression of *Pax6*(green) and *NK2.1* (red).

After investigating the evolutionary relationship of mushroom bodies of *Platynereis* and insects, the next question I addressed was: Is there any homology between vertebrate telencephalon and *Platynereis* forebrain (including mushroom body) at the molecular level? In vertebrates, the telencephalon primordium is specified by the expression of *BF1*(*Foxg1*, Forkhead box G1)(Hebert and Fishell, 2008; Tao and Lai, 1992) in the anterior neural plate. The interactions between several genes including *BF1*, *Shh*, *Gli3* and *FGFs* specify the dorso-ventral identities in telencephalon. Further in development, the telencephalon is subdivided into four major domains, marked by the partially overlapping expression domains of *Emx*, *Pax6*, *Gsx* and *Nk2.1* in dorsal to ventral direction(Hebert and Fishell, 2008). Moreover, many studies have shown that morphogens *Wnts* and *Bmps* are expressed dorsally in the telencephalon, and are essential for the proper patterning. Using WMISEP, firstly, I investigated the expression patterns of these telencephalon patterning genes in *Platynereis dumerilii* 48 hpf larvae to address the following questions: (1) Are *Gli* and *BF1* expressed in lateral to medial gradient, analogous to their dorso-ventral expression in vertebrate telencephalon? (2) Do the expression domains of *Emx*, *Pax6*, *Gsx* and *Nk2.1* co-localizes with that of *BF1*? (3) Are these genes also expressed in the same spatial order in *Platynereis* as in the vertebrate telencephalon?

As shown in the Figure 46A, *BF1* has relatively broad anterior expression domain in *Platynereis* larval brain. Further, I tested the spatial relationship of the expression domains of *Gli* and *BF1*. Indeed, *Gli* and *BF1* are expressed in the lateral to medial spatial gradients, similar to their dorso-ventral gradient in the vertebrate telencephalon. Furthermore, I investigated the co-expression of *Emx*, *Pax6*, *Gsx*, and *NK2.1* with *BF1*. Interestingly, *Emx*, *Pax6*, *Gsx* and *Nk2.1* are indeed co-expressed with *BF1*(shown in Figure 46B-F). Moreover, they are expressed in the lateral to medial gradients, similar to their dorso-ventral spatial expression in vertebrate telencephalon. Notably, the expression domain of *Emx* is relatively smaller in *Platynereis* in comparison to vertebrates. This is in agreement with the findings, which have suggested the expansion of *Emx* expression domain as a cause for the enlargement of vertebrate cortex. Further, I analyzed the expression of *Wnts* in *Platynereis*. Indeed, as shown in Figure 46H,I, I found *Wnt5*, *Wnt8*

and *WntA* to be expressed laterally in the larval brain. Furthermore, I investigated the co-expression of *Lhx2/9*, a LIM homeodomain gene, with *BF1*. Several studies (Ando et al., 2005; Bulchand et al., 2001; Hebert and Fishell, 2008; Mangale et al., 2008; Monuki et al., 2001; Porter et al., 1997) have shown that *Lhx2* is essential for the proper patterning of telencephalon. Indeed, *Lhx2/9* perfectly overlaps with the *BF1* expression domain (Figure 46G).

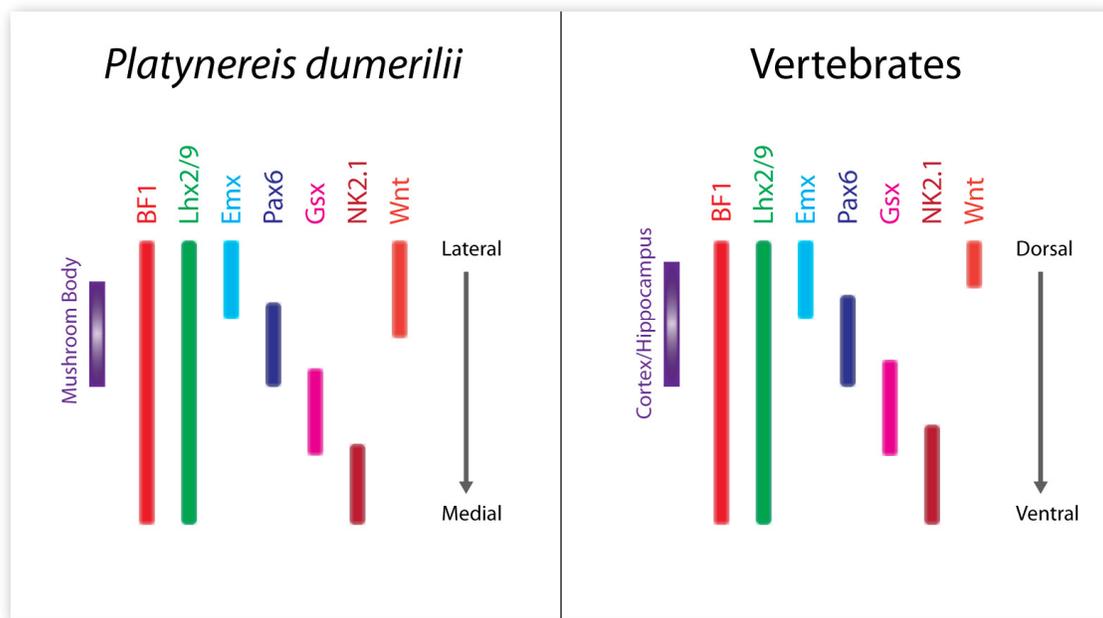


Figure 47. Schematic comparison of early telencephalon patterning genes in *Platynereis* and vertebrates.

Left panel shows the schematic of spatial organization of early telencephalon specification genes in *Platynereis dumerilii*, based on the data shown in Figure 46. Right panel summarizes the expression of early telencephalon specification genes in vertebrates (based on (Hebert and Fishell, 2008)). Comparing the two schematics suggests the striking similarities. Moreover, the mushroom bodies in *Platynereis* (Figure 40D) and cerebral cortex/hippocampus in vertebrates originates from similar locations in the molecular topography.

2.3.3 Common origins of telencephalon and Mushroom Body specifying gene network?

Vertebrate telencephalon is subdivided into dorsal (pallium) and ventral (subpallium) telencephalon. Dorsal telencephalon is specified by the overlapping expression gradients of *Emx* and *Pax6* while the ventral telencephalon is specified by *Gsx* and *NK2.1*. The

dorsal telencephalon is the major source of vertebrate cerebral cortex, which controls most of the complex behaviors of vertebrates, including memory formation and learning, by integrating the sensory information received from various sensory organs including olfactory and visual information. The corresponding molecular region (*Emx*, *Pax6* and *BFI*) in *Platynereis* brain gives rise to mushroom bodies (schematic summary in Figure 47), marked by the co-expression of *Dach*, *Pax6* and *BFI* (Figure 40D). Furthermore, the dorsal telencephalon mainly gives rise to glutamatergic neurons. As shown in the section 2.2.4, *Platynereis* mushroom bodies mainly possess glutamatergic neurons. Altogether, this investigation presents a striking example of the conserved gene battery involved in specifying the functionally equivalent brain structures in *Platynereis* and vertebrate.

3. DISCUSSION

3.1 Whole Mount In silico Expression Profiling for Evo-Devo research

3.1.1 *WMISEP protocol*

One of the many ways to investigate the evolution of sophisticated organs such as brain is to investigate the different cell types that make that organ. Cell types are defined by a unique combination of genes (molecular fingerprint, (Arendt, 2008)) that specify the distinct morphological and physiological features that are characteristic of that cell type. As a consequence, we are required to investigate the co-expression of several genes at cellular resolution to identify the cell types. The most common way to study the co-expression of genes is based on fluorescence whole mount in situ hybridization, which produces different color precipitates to mark the expression patterns of two genes at a time. Though this protocol has already been established in many model organism (including *Platynereis dumerilii*, (Tessmar-Raible et al., 2005)), a systematic screen for the co-expression of several genes is often very tedious and time consuming, owing to the fact that the weakly expressed genes do not produce good enough signal and the signal to noise ratio is generally low. Moreover, for instance, to investigate the co-expression of 20 genes we would need to do 190 ($^{20}C_2$) double fluorescence experiments and this number increases non-linearly ($n*(n-1)/2$, n = number of genes) as more and more genes are required to be investigated. Therefore, to facilitate such a high-throughput screen, I developed a protocol, the so-called Whole Mount In Silico Expression Profiling (WMISEP), combining advanced image processing algorithms, whole mount in situ hybridization, immunostaining against acetylated tubulin and the whole mount reflection microscopy (Jekely and Arendt, 2007). The basic idea of the protocol is to acquire two-channel confocal image stacks, with one channel containing expression information for the gene and the other channel containing the information of the axonal scaffold. The information in the axonal scaffold channel is then used to align several such images to a common reference average axonal scaffold image, and thus bringing the expression patterns into the same coordinate system. This protocol is very useful for doing fast high-throughput co-expression screens for several genes. For instance, to analyze the co-

expression of 20 genes we would need only 20 experiments, and moreover, to add an additional gene to the analysis would require only one extra experiment. Subsequently, I conducted several experiments to illustrate the cellular resolution sensitivity and specificity of the WMISEP protocol. I optimized the WMISEP protocol for several larval stages of *Platynereis dumerilii* development, including 48 hpf, 56 hpf and 4 dpf. WMISEP has been used to generate cell resolution expression of 72 genes. This data set can generate co-expression information for 2,556 (${}^{72}C_2$) double combinations, 59,640 (${}^{72}C_3$) triple combinations and so on. Furthermore, I developed a cellular model of the 48 hour old *Platynereis* larval brain that resulted in cellular gene expression profiles.

3.1.2 Clustering of larval brain cells and genes based on expression profiles and spatial patterns

The cellular expression profiles dataset provides a unique opportunity to investigate the relationships among the larval brain cells, based on their gene expression profiles and their spatial patterns in the brain, respectively. I used several clustering approaches to cluster the larval brain cells based on their expression profiles. Furthermore, I developed methods to visualize the spatial organization of larval brain cells that belong to a particular cluster. The overall clustering analysis of all the cells in the *Platynereis* larval brain resulted in three super clusters, which I termed ClusterA, ClusterB and ClusterC. Further analysis of the spatial organization of the cluster-specific cells revealed that ClusterA contains lateral cells, ClusterB contains medial cells and ClusterC contains cells in the stomadeum (Figure 26, 28). In summary, the biological messages from the cluster analysis are: Cells having similar expression profiles tend to aggregate together spatially; the lateral cells, belonging to ClusterA, are most likely to be the sensory organ cells whereas the medial cells, belonging to ClusterB, are possibly the sensory information integration system (the brain proper), controlling the behavior of *Platynereis* larvae; and last but not the least, the clustering analysis also provides information on the intermediate stages of cell type specification, as hitherto undifferentiated cells tend to cluster with their differentiated counterparts. Combining these clustering analyses data with the

developmental lineage history of the cells would further increase our understanding of the cell type specification processes in development.

Additionally, the cellular expression profile data also give us the opportunity to investigate the relationships among genes, based on their spatial expression patterns. I used the clustering techniques to identify the clusters of genes that are expressed in similar patterns. These clusters are most likely to represent the modules of gene regulatory networks involved in development. For instance, as shown in the Figure 30, the genes involved in eye development clustered together with significant statistical support from the data. Moreover, I could also identify an additional gene, *KLF* (Krueppel-like factor), which was not known to be expressed in the eyes hitherto. Also, I investigated some of the significant clusters, including *Chx10* and *Lhx2*; *Dbx* and *NK2.1*; *Wnt8* and *Pax6*, in further details. As shown in Figure 31, the expression domains of *Chx10* and *Lhx2* largely overlap, *Dbx* expression is almost exclusively in *NK2.1* expression domain and *Wnt8* is expressed mainly in *Pax6* expression domain. These analyses obviously suggest possible regulatory links between these genes, and hence facilitate better experimental designs for investigating gene regulatory networks. Another approach to gain insight into the regulatory networks involved in specifying a particular cell type is to compare the expression profiles of such cell type present in different spatial locations in the brain. For instance, the comparison of expression profiles of glutamatergic neurons that are found in several distinct locations in the brain would provide us the information about the common genes that are expressed in those cells, and thus possibly identifying the pathways specific for generating the glutamatergic phenotype. As more and more genes would be included into the cellular expression profiles, the better it will become for predicting real regulatory interactions.

Another interesting type of question that can be asked from the cellular expression profiles dataset is: How many distinct cell types are specified in the expression domain of a particular master regulator gene? I investigated this question with an example study of the *Dach* expression domain. Firstly, I assembled a collection of cells that are positive for *Dach* and clustered them based on their overall expression profiles. This resulted in three statistically significant super-clusters and several sub-clusters. The three super-clusters are clearly functionally distinct as they give rise to at least three different cell type

clusters, namely mushroom body cells, eye field cells and a hitherto unknown cluster. Similar analysis can be easily extended to several other master regulator genes, and such analysis would obviously become more robust as more genes would be put into the system.

3.1.3 Future perspectives for WMISEP

As evident from the results presented in this thesis, the WMISEP protocol can be easily used to address several different kinds of questions. Here I present two of the many possible future directions:

(I) To use WMISEP for studying the dynamics of co-expression of several genes to predict cell type specific gene regulatory networks. This would require the cellular resolution expression profiles of different, but not so different, temporal stages of development, and the generation of unambiguous developmental continuity among these stages. The temporal stages would have to be chosen wisely in such a way that the consecutive developmental stages could be easily connected morphologically. One of the many possible configurations for *Platynereis dumerilii* development stages would be the continuity of 40 hpf, 48 hpf and 56 hpf. These temporal stages of *Platynereis* larval development can be easily connected to each other using WMISEP and still possess enough interesting morphological and gene expression variability. As a proof of principle, I established WMISEP for 56 hpf *Platynereis* larvae, as shown in section 2.1.1.9, and generated the average expression data for several genes including *Ascl*, *BFI*, *Dach*, *Dbx*, *Dll*, *Emx*, *Ngn*, *Pax6*, *Six3*, *Syt*, *VACHT* and *VGluT* (Figure 22). To connect the two temporal stages of development, I aligned the average axonal scaffold of 56 hpf larval brain to that of 48 hpf larval brain. As shown in Figure 48, the average axonal scaffold of 56 hour old and 48 hour old larval brain appears largely similar but with some interesting differences (Figure 48, marked by arrows). For instance, some of the obvious differences are that there are 4 additional sensory organs, located superficially on each side, in 56 hpf brain but are absent in 48 hpf brain and the shape of basal-ventral axonal scaffold of 56 hpf is slightly compressed compared to 48 hpf (Figure 48). Subsequently, to investigate the gene expression variations, I transformed the average gene expression patterns of the 56 hpf larval brain reference system to the 48 hpf larval brain reference

system, using the same parameters that were used for mapping the 56 hpf average reference brain to 48 hpf average reference brain. As a result, the expression of genes in 56 hpf larvae can now be directly compared to their expression at 48 hpf, and therefore generate a developmental spatial and temporal continuity. It is evident that the expression domain of some genes is not changed over time (48 hpf to 56 hpf) but in some cases there is a significant alteration. For instance, *BFI* expression domain largely remains unchanged in 56 hpf compared to 48 hpf, the expression domain of *Dll* has expanded significantly, and the expression domains of *Ascl* and *Ng2* are specifically reduced. Another interesting example is that of the *Pax6*, *Dach* and *Emx* expression domains. As shown in the Figure 49, though *Pax6* and *Dach* expression is slightly expanded, there is a distinct hole appearing in their expression domain (marked in the Figure 49). Intriguingly, the *Emx* expression domain has moved ventrally to fit into that hole, implying that either the *Emx* expressing cells have moved ventrally or it is a wave movement of *Emx* expression and hence down-regulation of *Pax6* and *Dach* in those cells, later being less likely. Investigating the 52 hpf developmental stage would shed more light on this phenomenon. Also, another interesting example is that of *VGluT* (glutamatergic neurons marker) and *VACHT* (cholinergic neurons marker) expression domains, which have greatly expanded laterally and medially, respectively. Such examples provide us unique opportunities for reconstructing the gene regulatory network involved in specifying specific cell types (for example glutamatergic and cholinergic cell types). For instance, those spatial domains which express *VGluT* and *VACHT* in 56 hpf can be traced back to corresponding regions in 48 hpf. And a screen for the combinations of genes that are uniquely expressed in those regions in 48hpf, and not anywhere else, would yield the most likely candidates regulating those cell fates.

(II) To extend the WMISEP protocol to several other species for cell type comparisons. There is a paucity of experimental manipulation techniques in many of the non-conventional model systems used for evo-devo research. For comparing cell types across many species, we need methods to study co-expression of several genes, along with a morphological characterization. The WMISEP protocol could be adaptable to many different species, to generate molecular maps of cell types present in those species. As shown in this thesis, this protocol requires very basic experimental manipulation

techniques including whole mount in situ hybridization and immunostaining against proteins which provides a reasonable scaffold for alignment. It is worth establishing WMISEP in as many species as possible as it would provide us with an unprecedented high resolution and detailed molecular profiling of different cell types.

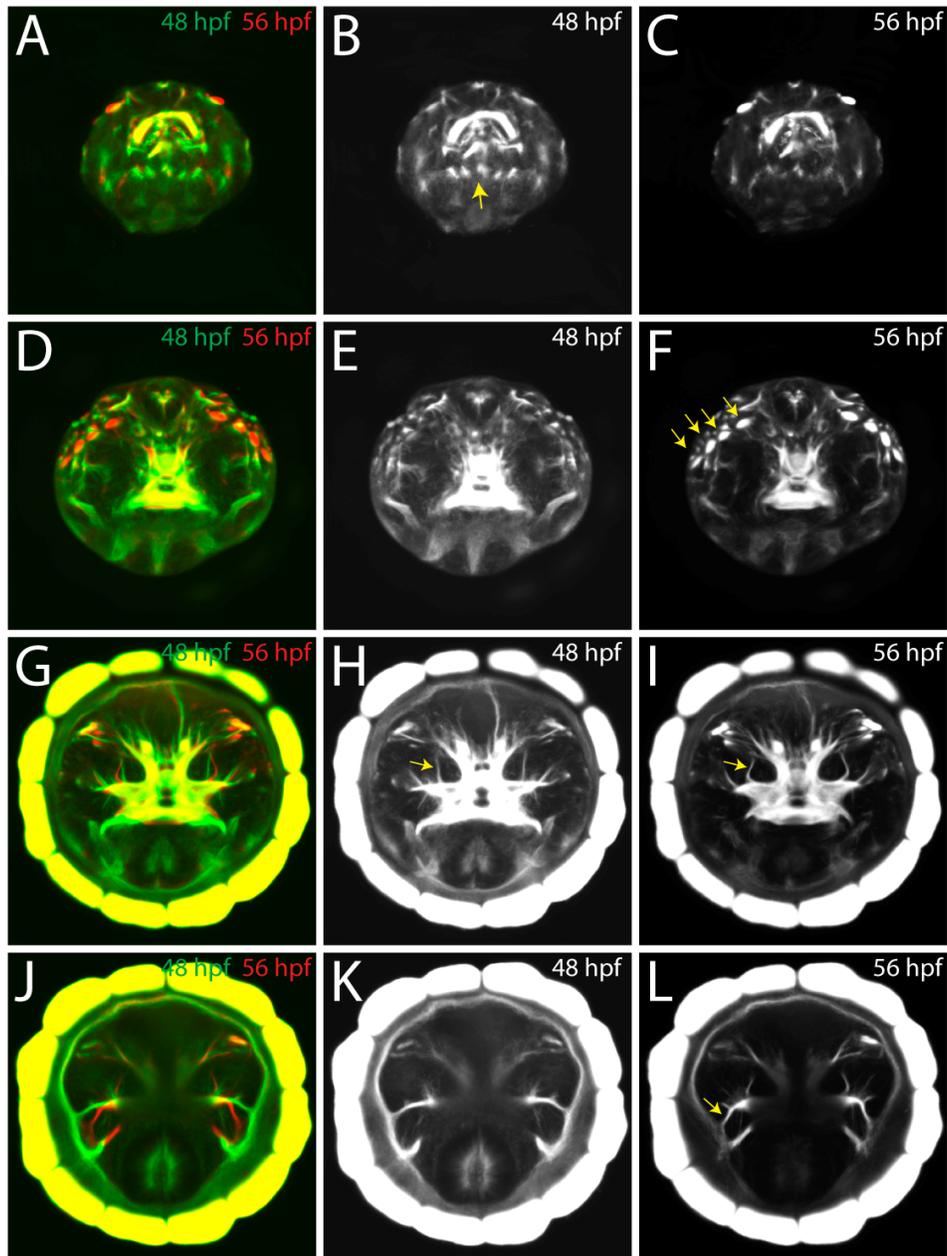


Figure 48. Direct comparison of average axonal scaffold of 48 hpf and 56 hpf *Platynereis* larval brains.

(A,D,G,J) show a series of optical sections, apical to basal, directly comparing the brain of 48 hpf (green) and 56 hpf *Platynereis* larvae. (B,E,H,K) show the series of optical sections of 48 hpf and (C,F,I,L) show the series of optical sections of 56 hpf larval brain. The yellow arrows illustrate some of the obvious

differences. For instance, there are 4 additional sensory structures, located superficially on each side, in 56 hpf brain but are absent in 48 hpf brain (marked in B), and the shape of basal-ventral brain of 56 hpf is a bit compressed compared to 48 hpf at the most basal level (shown in L).

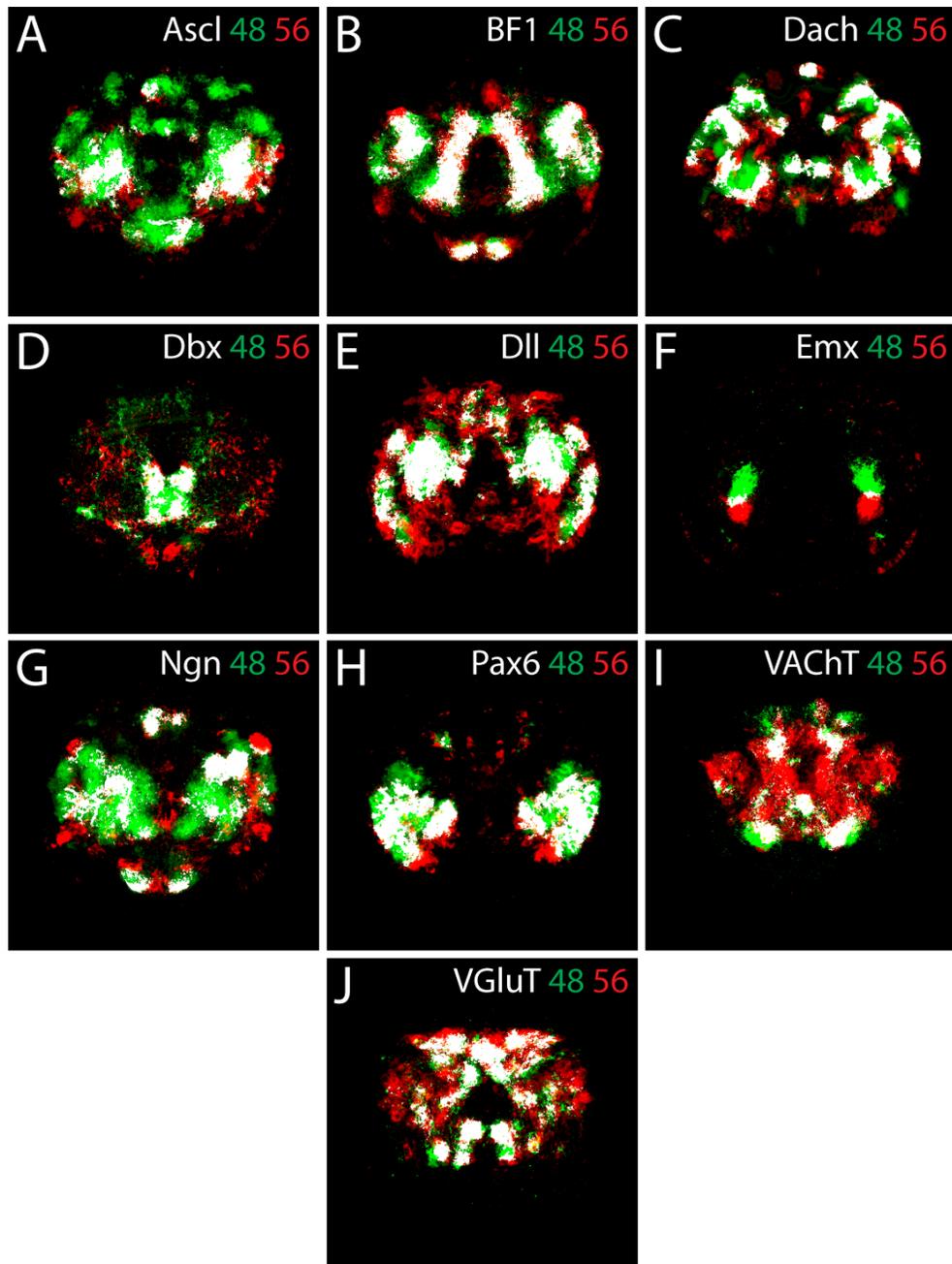


Figure 49. Direct comparison of gene expression in 56 hpf and 48 hpf *Platynereis* larvae.

Green represents the expression in 48 hpf and red represents the expression in 56 hpf. All images are in apical view. (A-J) shows the direct comparison of expression patterns of *Ascl*, *BF1*, *Dach*, *Dbx*, *Dll*, *Emx*, *Ngn*, *Pax6*, *VACHT* and *VGLuT* in 48 hpf (green) and 56 hpf (red).

3.2 Evolution of mushroom body and telencephalic cell types

3.2.1 Evolution of insect and annelid mushroom bodies

The mushroom bodies were first discovered in 1850 by Dujardin, and since then they have been a major topic in neuroanatomy and behavioral research. Mushroom bodies are prominent lobed neuropils made of parallel axon bundles, supplied by clusters of small-diameter globuli cells located dorsally in the anterior part of the brain. Similar structures have been described in many arthropods and annelids (Strausfeld et al., 1998). Ever since their discovery, the mushroom bodies have been considered to be bestowing the intelligent behavior to various species. Many studies have indeed found evidence of their involvement in higher-order brain functions. For instance, (Vowles, 1964) showed that ants with perturbed mushroom bodies lost their ability to negotiate a maze using olfactory cues. Several experiments done in honey bee further demonstrated the correlation between behavioral complexity and mushroom body size. For instance, the queens and the workers of honey bees possess bigger mushroom bodies compared to drones (Alten, 1910; Forel, 1874; Jonescu, 1909). Furthermore, in cockroaches also the mushroom body has been shown to play a role in place memory formation (Mizunami et al., 1998b). Moreover, several detailed investigations in *Drosophila* have pointed out a role of mushroom bodies in associative learning and memory formation (de Belle and Heisenberg, 1994; Han et al., 1992; Heisenberg et al., 1985), and recently in controlling their sleep-like behavior (Joiner et al., 2006; Pitman et al., 2006). In essence, the mushroom bodies can be considered as sensory information integration units. The types of sensory information arriving at the mushroom bodies tend to vary in distantly related species and hence their roles in controlling different behaviors. For instance, in the honey bee mushroom bodies also receive visual input in addition to the olfactory input, which is in contrast to *Drosophila*. However, even after so many detailed investigations, the evolutionary origins of mushroom body remain controversial. The current understanding on the mushroom body evolution is summarized by (Farris, 2005) as “*Considering the lack of mushroom bodies in the most basal hexapod lineages and assuming that*

crustacean hemiellipsoid bodies are not homologous to mushroom bodies, it seems likely that mushroom bodies arose independently more than once in the invertebrates.” The fact that mushroom bodies are not found in crustaceans and basal hexapods raises the possibility of mushroom bodies having evolved independently many times. However, crustaceans possess structures called hemiellipsoid bodies which resemble mushroom bodies but without proper pedunculi, and therefore can also be considered highly derived mushroom bodies. Moreover, mushroom body-like structures have been identified in many annelid species (e.g. *Platynereis dumerilii*, *Nereis diversicolor*) as well. Most of the studies comparing mushroom bodies across many species have been at gross anatomical and histological level. There is a need for looking at the molecular details of mushroom body specification networks in different species to shed more light on their evolution.

In my thesis work, I aimed to look at the molecular fingerprint of mushroom body cell types in *Platynereis dumerilii* and compare it with the cell types in mushroom bodies in insects (*Drosophila* mainly). Firstly, I carried out detailed investigations of the structure of mushroom bodies in *Platynereis*. As shown in the section 2.2.2, *Platynereis* mushroom bodies are composed of thousands of cells that send parallel axons posteriorly to form a stalk-like peduncle which further gives rise to medial lobes. Furthermore, I found that mushroom bodies receive connections from several sensory organs including palps (presumed chemosensory organs) and generate output via the medial lobes. In terms of structural comparison, it is clear that the annelid mushroom bodies are very similar to insects mushroom bodies, as suggested before (Strausfeld et al., 1998). Subsequently, I aimed to investigate the development of mushroom bodies to trace their embryonic origins. By investigating a number of intermediate larval stages, I could reliably trace back the mushroom body development to 3 days old larvae. Further, I screened several candidate genes to identify markers that are expressed in the mushroom bodies. As a result, I found that both *Dach* and *Pax6* are co-expressed in *Platynereis* mushroom bodies. Using the combination of anatomical details and molecular markers, I could identify the mushroom body anlagen at embryonic stages. In summary, it is evident that the mushroom body originates from few cells located ventrally and laterally in 24 hour old embryos (**Figure**). Subsequently these mushroom body anlagen cells proliferate and move slightly medially in 2-days-old larvae. These cells are still undifferentiated as

indicated by the lack of expression of *synaptotagmin*. By 56 hour post fertilization, the mushroom body Anlagen cells are slightly scattered and by 3 days, they already have two distinct (dorsal and ventral) pedunculi on each side. Thereafter, the mushroom bodies proliferate in the same location throughout.

After having identified the mushroom bodies Anlagen in earlier larvae stages, I used WMISEP to screen for genes that are expressed in these cells, to define the molecular fingerprint. Furthermore, I did a thorough literature search to define the molecular fingerprint for insect mushroom bodies. As shown in the section 2.2.5, the comparison of *Platynereis* and insect mushroom body molecular fingerprint suggests their deep homology, indicating that probably the last common ancestor of ecdysozoans and lophotrochozoans already possessed mushroom body like cell types.

Platynereis shows a relatively complex behavior (Evans, 1966a; Evans, 1966b; Evans, 1986). What is the function of large mushroom bodies in *Platynereis*? For addressing this question, I (together with Julien Colombelli, Stelzer Group, EMBL Heidelberg) established a method to ablate the mushroom bodies in living worms using LASER based surgery. I showed that the LASER induced damage is indeed specific to mushroom bodies. Further work is needed to establish robust behavioral assays for identifying the function of *Platynereis* mushroom bodies.

3.2.2 Homology of mushroom body and dorsal telencephalon cell types

The next question I asked was: Is there any homology between mushroom body and vertebrate telencephalic cell types? Since their discovery in 1850, several comparisons have been drawn between mushroom bodies and various vertebrate brain centers. Dujardin himself compared them to folds and gyri of the cerebral cortex. Further, (Hanström, 1928) suggested the analogy of mushroom bodies and the vertebrate thalamus. Furthermore, the mushroom bodies have been compared to vertebrate hippocampus (Kandel and Abel, 1995; Mizunami et al., 1998b). From functional perspectives, the mushroom bodies are essentially involved in similar functions as the vertebrate hippocampus/cortex. The most common view is that these functional

similarities are the result of convergent evolution (Farris, 2005; Strausfeld et al., 1998), owing to similar ecological pressures. On the other hand, the evolutionary origin of the telencephalon, which gives rise to the cerebral cortex and the hippocampus, is also enigmatic. Supposedly, the absence of telencephalon-like structure in the basal chordate amphioxus has indicated that the telencephalon likely arose only after the evolutionary split of amphioxus from the vertebrate stem line (Murakami et al., 2005). However, it is also possible that amphioxus lost the telencephalon secondarily, or that a telencephalon precursor has so far escaped notice. To address the evolutionary relationship of telencephalon and mushroom body further, I aimed to compare the expression of early telencephalon development gene batteries in developing *Platynereis* larval brain and vertebrates. As revealed by detailed investigations in mouse and fish models, the telencephalon primordium is specified by the expression of *BF1* (*Foxg1*, Forkhead box G1) (Shimamura and Rubenstein, 1997; Tao and Lai, 1992) in the anterior neural plate. The interactions between several genes including *BF1*, *Shh*, *Gli3* and *FGFs* specify the dorso-ventral identities in telencephalon. Later in development, the telencephalon is subdivided into four major domains, marked by the partially overlapping expression domains of *Emx*, *Pax6*, *Gsx* and *Nk2.1* in dorsal to ventral sequence (Hebert and Fishell, 2008). Moreover, many studies have shown that *Wnts* and *Bmps* are expressed dorsally in the telencephalon, and are essential for proper patterning (Hebert and Fishell, 2008). Firstly, I used WMISEP to investigate the expression patterns of these regionalization genes in the developing *Platynereis* larval brain to test if there is any similarity in the spatial distribution of expression. As shown in section 2.3.1, *BF1* and *Gli* are expressed indeed in lateral-medial orientation corresponding to the dorsal-ventral orientation within the vertebrate neural tube. Furthermore, I found that *Emx*, *Pax6*, *Gsx* and *NK2.1* are indeed co-expressed with *BF1* in similar spatial orientation. I also showed that *Lhx2/9*, a LIM homeodomain gene, and *Wnt5* and *Wnt8* are co-expressed with *BF1*, similar to their expression in the telencephalon. Moreover, *Emx* and *Pax6* expression domains in mouse telencephalon give rise to cerebral cortex and hippocampus. Intriguingly, the mushroom body anlagen cells in *Platynereis* larval brain are located in an equivalent position within the *BF1* expression domain. Notably, though amphioxus does not have recognizable telencephalon, many of these telencephalon patterning genes are expressed in cerebral

vesicle (which is often compared to diencephalon instead of telencephalon)(Benito-Gutierrez, 2006; Murakami et al., 2005). Thus, an obvious question to ask would be: What is the significance of this striking similarity between these early expression patterns? The most likely explanation is that these genes are part of an ancient gene regulatory network that was already present in the last common ancestor of all bilaterian animals. In the terminology of Eric Davidson, this ancient gene regulatory network would represent a pan-bilaterian kernel(Davidson and Erwin, 2006), and the distinct lineages of vertebrates and invertebrates acquired specific “plug-in” gene regulatory networks to generate the present diversity. For instance, it has been proposed that the acquisition of *reelin* signaling in vertebrates led to the expansion of the cerebral cortex (Nomura et al., 2008). At the cell type level, it is likely that the cell types similar to mushroom body and telencephalon cells already existed in Urbilateria (Figure 50). Additional functional studies are required to test this hypothesis.

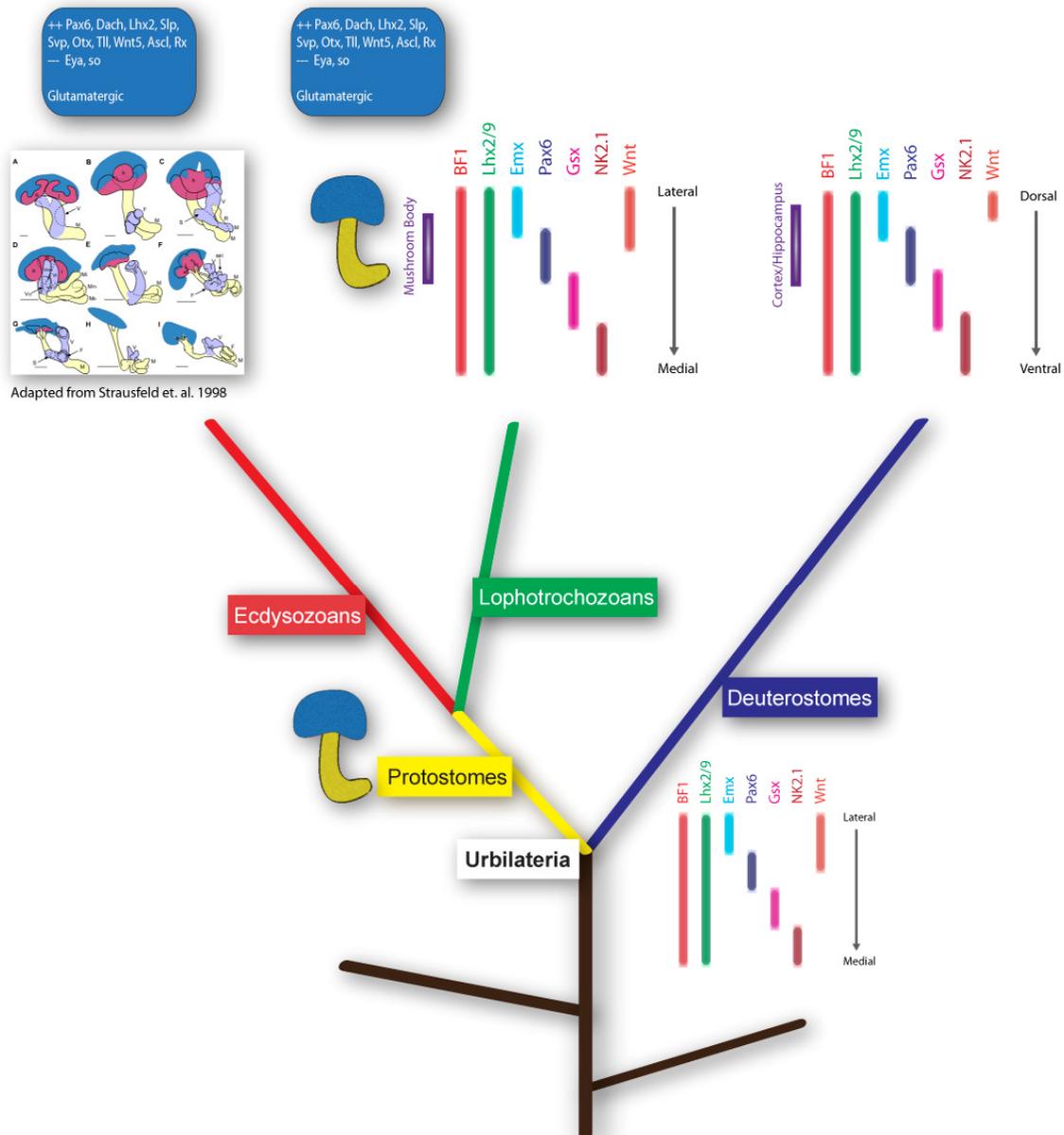


Figure 50. Schematic summary of evolution of mushroom body and telencephalon.

The comparison of mushroom body and telencephalic cell types in bilaterian animals. The lineages of three superphyla Ecdysozoa, Lophotrochozoa and Deuterostome, are represented by red, green and blue lines respectively. Comparison of the mushroom body anatomy and molecular fingerprint suggest that the last common ancestor of Ecdysozoa and Lophotrochozoan already possessed mushroom body like cell types. Moreover, the comparison of telencephalon patterning genes in vertebrates and *Platynereis* raises the possibility that Urbilateria already possessed this gene regulatory network. Additionally, the mushroom body and cerebral cortex/hippocampus cell types originate from similar coordinates in the conserved molecular topography.

4. Materials and Methods

4.1 Experimental materials and methods

4.1.1 *Platynereis dumerilii* culture

Our laboratory at EMBL, Heidelberg maintains a stable culture of *Platynereis dumerilii*. The animals are kept at 18°C (as described by (Fischer and Dorrestijn, 2004)) with an artificial moon cycle (1 week of artificial moonlight and 3 weeks of darkness). All the animals and embryos used during my studies come from this culture.

4.1.2 cDNA templates for degenerate PCR and 3',5' RACE reactions

For cloning the genes using degenerate PCR and 5',3'-RACE extensions, I used the cDNA templates generated using the SMART technology (SMART RACE cDNA amplification kit, Clontech). The total RNA was extracted from the larvae at various developmental stages using RNAeasy kit (Qiagen) or the Trizol extraction method. The standard protocol from the SMART manual was followed to generate the cDNA libraries.

4.1.3 Cloning of *Dachshund* (*Dach*)

Before I started my thesis, Maria Inês Medeiros de Campos Baptista (Diploma thesis, EMBL Heidelberg and University of Lisbon, 2002) had already attempted to clone *Dach* from *Platynereis dumerilii*. She managed to get a short fragment (84 base pairs long) using degenerate PCR, but the RACE extension product remained elusive. Moreover, when I started, the physical clone for the fragment was not available to me and thus I set out to start from scratch. Firstly, I used a PCR reaction using degenerate primers (designed in the evolutionarily conserved regions) to amplify the short fragment of *Dach*.

The primers used are as follow:

- 1) Dac_Deg_Up0: 5' ATH TGY YTN CCN CAR GCI TTY GA 3'

- 2) Dac_Deg_Lo2: 5' TNA CNC CNG GYT GDA TNG CNC C 3'
- 3) Dac_Deg_Lo3: 5' NMT RTA NAR NGT YTC RAA RTC YTT 3'

The following combinations of primers were used:

1. Dac_Deg_Up0 and Dac_Deg_Lo2, Annealing temperature used: 48 °C
2. Dac_Deg_Up0 and Dac_Deg_Lo3, Annealing temperature used: 44 °C
3. Nested: Dac_Deg_Up0 and Dac_Deg_Lo2 with the PCR product of reaction (2) as template, Annealing temperature used: 48 °C

For each 50 µl reaction:

Primer 1 (5 µM)	: 5 µl
Primer 2 (5 µM)	: 5 µl
10X Buffer	: 5 µl
Template	: 1 µl
Taq Polymerase	: 1 Unit
H ₂ O	: up to 50 µl

PCR program used:

1. 95 °C for 2 mins
2. 95 °C for 1 min, Annealing Temperature for 2 min, 72 °C for 1 min
3. Repeat step 2, 35 times.
4. 72 °C for 10 minutes.

This reaction yielded an 84 bp fragment of *Platynereis Dach* gene. Subsequently, I carried out 3' and 5' RACE extensions for getting a bigger fragment of the gene. The primers were designed in the middle of the fragment.

The primers used:

- 1) DacRACE_F1 = TACGGTGTACACCAAACCTCAAAGACTG
- 2) DacRACE_F2 = CAAAAGACTGGACATTACTCCCATCGTG
- 3) DacRACE_B1 = GAGGATCCGGACCTGCTCCACATTGCAC

4) DacRACE_B2 = CTGCTCCACATTGCACACGATAGGAGTA

The following combinations of primers were used:

1. DacRACE_F1 and UPM,
2. DacRACE_F2 and UPM
3. DacRACE_B1 and UPM
4. DacRACE_B2 and UPM
5. Nested: DacRACE_F2 & UPM and DacRACE_B2 & UPM

Note. UPM is Universal Primer Mix from SMART RACE kit, Clontech

For each 50 µl reaction:

Primer 1 (5 µM)	: 5 µl
Primer 2 (5 µM)	: 5 µl
10X Buffer	: 5 µl
Template	: 1 µl
Taq Polymerase	: 1 Unit
H ₂ O	: up to 50 µl

PCR program used:

1. 95 °C for 2 mins
2. 95 °C for 1 min, Annealing Temperature (58°C) for 2 min, 72°C for 2 min
3. Repeat step 2, 35 times.
4. 72 °C for 10 minutes.

The combined 3' and 5' RACE extensions yielded 1,755 base pairs of *Dach*.

The sequence and the neighbour-joining phylogenetic tree are reported in Appendix.

4.1.4 Cloning of *Svp/COUPTF1*

I designed degenerate primers in the evolutionarily conserved regions (assessed by multiple sequence alignment, using ClustalW, of the protein sequences from several species) of *svp/COUPTF1*.

The primers used:

- 1) COUPTF1_U1: ATHGARTGYGTNGTNTGYGGNGA
- 2) COUPTF1_U2: GYATGCARCCNAAYAAAYATHATGG
- 3) COUPTF1_L1: GTYTCDATNGGNGTYTTNCCNAC
- 4) COUPTF1_L2: CKNGTNGGGYTGRRTNGGRTAYTG
- 5) COUPTF1_L3: GCRCAYTGNGCYTTYTCYTG
- 6) COUPTF1_L4: TTNARYTTYTCNACYTGYTCYTG

The following combinations of the primers were used:

COUPTF1_U1 and COUPTF1_L1, Annealing temperature: 59 °C

COUPTF1_U1 and COUPTF1_L2, Annealing temperature: 59 °C

COUPTF1_U1 and COUPTF1_L3, Annealing temperature: 59 °C

COUPTF1_U1 and COUPTF1_L4, Annealing temperature: 57 °C

COUPTF1_U2 and COUPTF1_L1, Annealing temperature: 57 °C

COUPTF1_U2 and COUPTF1_L2, Annealing temperature: 57 °C

COUPTF1_U2 and COUPTF1_L3, Annealing temperature: 57 °C

COUPTF1_U2 and COUPTF1_L4, Annealing temperature: 57 °C

For each 50 µl reaction:

Primer 1 (5 µM)	: 5 µl
Primer 2 (5 µM)	: 5 µl
10X Buffer	: 5 µl
Template	: 1 µl
Taq Polymerase	: 1 Unit
H ₂ O	: up to 50 µl

PCR program used:

1. 95 °C for 2 mins
2. 95 °C for 1 min, Annealing Temperature for 2 min, 72°C for 1 min
3. Repeat step 2, 35 times.
4. 72 °C for 10 minutes.

This yielded in a 907 bp fragment of *Platynereis Syp* gene. The sequence and the phylogenetic tree are reported in Appendix.

4.1.5 Preparation of RNA probes for WMISH

I used the following procedure for preparing Dig- and Flu-UTP labeled RNA probes for whole mount situ hybridization.

Reagents:

NTP-Mix: ATP, CTP, GTP 15.4 mm each, UTP 10.0 mm (all Boehringer)

Dig-11-UTP 10 mm (Boehringer)

Fluo-12-UTP 10 mm (Boehringer)

T7-/SP6 -RNA-Polymerase 20 U/μl (Boehringer)

10xTranscriptionbuffer (Stratagene)

DNaseI RNase-free 10U/μl (Boehringer)

- linearize 10 μg of template with a suitable enzyme
- purify template from the enzyme and digestion buffer (using Illustra GFX PCR DNA and Gel Band Purification Kit, GE Healthcare)
- add in the following order to a total volume of 20 μl:

linearized template	1 μg
100 mm DTT	2 μl
NTP-Mix	1.3 μl
10 mm Dig-UTP/Fluo-UTP	0.7 μl
RNase inhibitor	0.5 μl
10xTranscriptionbuffer	2 μl

H ₂ O	add upto 19 μ l
RNA-Polymerase	1 μ l

- incubate for 2.5 hrs at 37°C
- add 1 μ l DNaseI and incubate for another 30 min at 37°C
- purify RNA using the Qiagen RNAeasy kit
- take an aliquot of 1 μ l and load in formamide loading buffer onto a TAE agarose gel to test the quality of the RNA probe
- dilute the remaining probe in 150 μ l Hyb-buffer and store at -20°C

4.1.6 WMISH combined with immunostaining against acetylated tubulin

Embryos Fixation

Reagents:

PTW - 1x PBS (pH 7.5), 0.1% Tween20 and sterile filtered (using 0.2 μ m, nitrocellulose).

PFA – 16% PFA/PBS stock

- fix embryos/larvae in 4% PFA/PTW (with shaking) for approximately 2.5 hours
- wash larvae 3 x 5 min in 1ml 2x PTW
- wash 5 min at room temperature in 100% MeOH
- replace MeOH and store embryos at least over night at -20°C

Rehydration and ProtK digestion

- transfer embryos to the nets placed in ~30 ml of 100% MeOH in the tip-box lids
- rehydrate 5 min in 75% MeOH/PTW
- rehydrate 5 min in 50% MeOH/PTW
- rehydrate 5 min in 25% MeOH/PTW
- rinse 2 x 5 min each in PTW

- digest the embryos with ProteinaseK (100 µg/ml PTW) without shaking for several minutes depending on the stage of the embryos. For 24 hpf to 72 hpf old larvae: 1 minute of ProtK digestion, for 72 hpf to 1 week old larve: 2 mins, for older than 1 week old larve: 3 min ProtK digestion.
- rinse 2x shortly in freshly prepared 2 mg/ml glycine/PTW
- fix in 4% PFA/PTW for 20 min
- wash 5 x 5 min in PTW

Hybridization

Reagents:

Heparin: make a stock of 50 mg/ml in H₂O, store at -20°C

Hybridization Mix: 50% formamide (Fluka, ultra pure), 5xSSC, 50 µg/ml heparin, 0.1%Tween20, 5 mg/ml torula RNA, store at -20°C,

For 50 ml of Hyb-Mix:

	<u>stock</u>	<u>Hyb-mix</u>
Formamide	100 %	25 ml
SSC	20 x	12.5 ml
Heparin	50 mg/m	150 µl
Torula-RNA (Sigma)	solid	250 mg
Tween20	10 %	500 µl
H ₂ O		add to 50 ml

- Transfer embryos to 1.5 ml Eppendorf tubes
- Pre-hybridize 1-2 hrs in 1 ml Hyb-Mix at 65°C
- Denature the RNA probe probe (5-15 µl in 200µl Hyb-Mix) in 250 µl of Hyb-Mix for 10 min at 80°C
- Remove pre-hybridization solution leaving embryos slightly covered to avoid their desiccation
- Quickly add the hybridization probe, mix gently and hybridize at 65°C overnight

Washes

Reagents:

4xSSCT: dilute 20xSSC to 4xSSC and add Tween20 to 0.2%

All steps are performed in a water bath, all wash solutions are pre-warmed to 65°C

- wash embryos 2 x 30 min in 1 ml 50% formamide/2xSSCT at 65°C
- wash embryos 15 min in 1 ml 2xSSCT at 65°C
- wash embryos 2 x each 30 min in 1 ml 0.2xSSCT at 65°C

Primary Staining

- block the embryos 1 to 2 hrs with 1ml of 5% sheep serum/PTW at room temperature with shaking
- incubate the embryos overnight at 4 °C, in 200 µl pre-absorbed anti-Dig-AP Fab (Roche) fragments (in 5% Sheep Serum/PTW) at a 1 : 2000 dilution and anti-acetylated tubulin (Sigma Cat. No. T6793) antibody at 1:250 dilution.
- wash 6 x 10 min with shaking in PTW at room temperature
- Make Staining Buffer (SB) in the mean time
- Equilibrate 2 x 5 min in Staining buffer
- Transfer the embryos to 24 well plate or 6 well plates
- dissolve 4.5 µl NBT (final 337.5 µg/ml) and 3.5 µl BCIP (final 175 µg/ml) in SB and add to the embryos
- Stain in the dark without shaking. Depending on the gene, leave up to few days. (Note: For Reflection microscopy, try to stain as strong as possible without getting much background to maximize the signal)
- wash 3 x 5 min in PTW

Secondary Staining

- Incubate the embryos overnight at 4 °C (in dark), in 200 µl pre-absorbed anti-mouse-FITC (Jackson Immunoresearch) (in 5% Sheep Serum/PTW) at 1:250 dilution and DAPI at 1 µg/ml final concentration.
- wash 6 x 10 min with shaking in PTW at room temperature

Mounting for imaging

- transfer the embryos to 87% glycerol (with 2.5 mg/ml DABCO)
- shake in dark for approximately 1 hour at room temperature (300 rpm)
- Store at 4 °C in Dark
- Mount on slides containing 3 stripes of tape for 48 hpf apical view confocal scanning.

4.1.7 Double fluorescence WMISH

The procedures for rehydration and hybridization are similar to the WMISH procedure as above. Two differently labeled probes (Dig-UTP and Flu-UTP labeled) were put together into the hybridization mix. Also, in place of anti-Dig-AP, anti-Dig-POD (Roche, Cat. No. 1 207 733) and anti-Flu-POD (Roche, Cat. No. 1 426 346) were used sequentially. For detecting the expressions of two genes, Perkin Elmer Cyanine 3 (Red precipitate) and Fluorescein (Green) TSA Plus System was used.

TNT: 0,1M Tris-Hcl pH 7,5; 0,15M NaCl; 0,1% Tween20

Primary staining

- block embryos 1(-2) hrs with 1ml of 1% Perkin Elmer Blocking Reagent/TNT at room temperature
- incubate embryos for 1(-2) hrs in 200 µl preabsorbed anti-Fluo-POD Fab fragments at a 1 : 50 dilution in 1%Blocking reagent/TNT overnight at 4°C (cover with aluminum foil)
- wash 6x 5' in TNT (make TNT buffer 1X stock. In the last wash transfer embryos from Nets to 2 ml eppendorfs)
- equilibrate 1x in 100ul TSA Plus Amplification Diluent
- dilute Fluorescein Fluorophore Tyramide 1:25 in TSA Plus Amplification Diluent (staining solution)
- add staining solution: 25ul/tube (this step onwards, always cover with aluminium foil)
- stain in the dark without shaking for 2h-5h
- check staining by transferring a few embryos in 3ml TNT in a 6-well plate; wash once with TNT, mount and have a look under the microscope
- wash 3x in TNT (can do one wash o/n in cold room)

POD (Peroxidase) enzyme inactivation

- incubate 20' in the dark in 1% H_2O_2 /TNT without shaking
- wash 4x 5' in TNT

Secondary staining

- block embryos 1(-2) hrs with 1ml of 1% Perkin Elmer Blocking Reagent/TNT at room temperature
- incubate embryos for 1(-2) hrs in 200 μ l preabsorbed anti-Dig-POD Fab fragments at a 1 : 100 dilution in 1%Blocking reagent/TNT overnight at 4°C; add 1 μ g/ml DAPI to antibody solution if desired
- wash 6x 5' in TNT
- equilibrate 1x in 100ul TSA Plus Amplification Diluent
- dilute Cy3 Fluorophore Tyramide 1:25 in TSA Plus Amplification Diluent
- add staining solution: 25ul/tube
- stain in the dark without shaking for 2h-5h
- check staining by transferring a few embryos in 3ml TNT in a 6-well plate; wash once with TNT, mount and have a look under the microscope
- wash 3x in TNT

4.1.8 Bright field and Confocal microscopy

I used Zeiss Axiophot microscope, with DIC optics, for acquiring bright field images using an appropriate objective (10X, 20X or 40X) depending on the developmental stage and the resolution desired. Often I took images at several different optical depths to visualize the expression patterns in three-dimension. For confocal microscopy, I used Leica TCS SP2 and Leica TCS SPE with 10X and oil emersion 40X objectives. Most of the images used for WMISEP protocol were acquired using Leica TCS SPE with 40X oil immersion objective. Depending on the fluorophore used for staining the embryos, appropriate LASER lines were used. For imaging the expression patterns visualized by NBT/BCIP precipitate, I used reflection microscopy settings established by (Jekely and Arendt, 2007). Typically I used the axial resolution of 1 micron and the horizontal resolution of 0.3756 microns.

4.1.9 BrdU assay

BrdU: use 10mM BrdU final concentration in natural sea water.

- ➔ from a 400mM stock solution: add 25ul/ml sea water
- incubate over desired time period
- fix embryos in 4%PFA/PTW for 2h
- transfer to 100%MeOH

BrdU detection

- rehydrate in 75%, 50%, 25% MeOH/PTW
- wash 2x PTW
- digest for 1 minute with Proteinase K (for embryos 48-72hpf)
- wash 2x with glycine/PTW (2 mg/ml)
- fix for 20' in 4% PFA/PTW
- wash 3x with PTW
- rinse 1x in ddH₂O / 0,1%Tween20
- incubate for 1hour in 2N Hcl / 0,1% Tween20 to denture the DNA
- rinse 4x with PTW
- Block embryos 15 minutes at RT in 2,5%sheep serum/1%BSA/PTW
- Block α -BrdU (clone BMC 9318, Roche) antibody 1:100 in the same blocking buffer for 15min
- Remove blocking buffer from embryos and add α -BrdU antibody
- Incubate 6x5min with shaking at RT
- Wash 6x5min with PTW
- Block antibodies: α -mouse-AP (Zymed Laboratories) 1:500 in 2,5%sheep serum / 1%BSA / PTW for 1h at RT
- Add DAPI to the α -mouse antibody
- Incubate overnight at 4 °C
- Wash 6x5' with PTW

- wash 2x in stain buffer (same as WMISH SB) and proceed to NBT/BCIP staining. The precipitation happens immediately
- Fix for 20 minutes in 4% PFA/PTW

4.1.10 Cryosections

- Anesthize the adult *Platynereis* worms in 50:50 7.5M MgCl₂:Natural Sea Water
- Fix the worms for 20' in 4% PFA/PTW
- Wash 5x 5 minutes in PTW
- Incubate for two hours in 15% Sucrose/PBS
- Overnight incubation in 30% sucrose/PBS at 4 °Celsius
- Mount in TissueTek (O.C.T. Compound) in the moulds and put on the Dry Ice
- Store at -20 °C
- Leica Cryostat was used for 8-40 microns thick sections. The sections were collected on cryosection optimized coated glass slides (SuperFrost Plus, MENZEL-GLASER)

4.1.11 Immunostaining on cryosections

- Rinse the slides 2X5 minutes in PTW in staining rack
- For the whole staining procedure it is advisable to use a wet box (by keeping wet tissues inside)
- Block for 30 minutes with 5% Sheep Serum/PTW at RT (use approximately 300 ul)
- Cover the sections with parafilm (**Ref**) stripes to prevent the evaporation
- Remove the parafilm, let the blocking solution drip off slowly (never let the sections dry out)
- Pipette 100 ul diluted primary antibody (in 5% S.S/PTW, 1:250 dilution for anti-acetylated tubulin) on to the slide and cover again with parafilm strips
- Incubate 1h at RT or overnight at 4 °C in the wet box
- Rinse slides 3X5 minutes in PTW in staining rack

- Pipette 100 ul diluted secondary antibody (1:250 dilution of anti-mouse FITC) on to the slide and cover with parafilm
- Incubate 1hour at RT or overnight at 4 °C in the wet box
- Rinse the slides 3X5 min in PTW in staining rack
- Mount with Moviol(**Ref**) and store at 4 °C

4.2 Computational material and methods

4.2.1 Computer Hardware

Most of the WMISEP related programs were run on EMBL Compute Cluster and a dedicated 8 processor Unix server, which was named evodevo. The evodevo server was bought from Dell with the configuration: Dell 2 X Quad-Core Xeon X5355 2.66 GHz / 2 X 4 MB Cache, 4 GB FB 667 MHz Memory (2 X 2 GB dual rank DIMMs). The server is housed in the central computing facility at EMBL and was initially installed by Michael Wahlers of IT Services at EMBL. I further configured the server for specific requirements of WMISEP and database. Additionally, a dedicated Windows 64bit XP machine was used for most of the image processing, Matlab and WMISEP expression profiles clustering related programming.

4.2.2 Softwares

Sequence analysis programs: BLAST (Altschul et al., 1990; Altschul et al., 1997), ClustalW (Thompson et al., 1994), ClustalX (Thompson et al., 1997), BioEdit (<http://www.mbio.ncsu.edu/BioEdit/BioEdit.html>)

Image Analysis, three dimensional Reconstruction and Modelling Softwares: ImageJ (<http://rsbweb.nih.gov/ij/>), Imaris (Bitplane AG, <http://www.bitplane.com/>), Amira (Visage Imaging Inc, <http://www.amiravis.com>)

Programming languages: C++, Java (Sun Microsystems), Perl (developed by Larry Wall), Matlab (<http://www.mathworks.com/>), R (<http://www.r-project.org/>)

Image processing libraries and binaries: DIPlib ver 1.6 (open source scientific image processing library of routines written in C, <http://www.diplib.org>), ITK 3.6 (Insight Segmentation and Registration Toolkit, <http://www.itk.org>), Elastix 3.9 (<http://elastix.isi.uu.nl/>), Torsten Rohlfing's multi-processor implementations of registration algorithms (Jefferis et al., 2007; Rohlfing and Maurer, 2003; Rohlfing et al., 2005).

Other Softwares: Eclipse IDE (Editor, Java programming, <http://www.eclipse.org>), MS Visual C++ (Editor for C++), VI (text editor in UNIX), Adobe Photoshop & Illustrator, MS Office

4.2.3 Transcription Factors screen

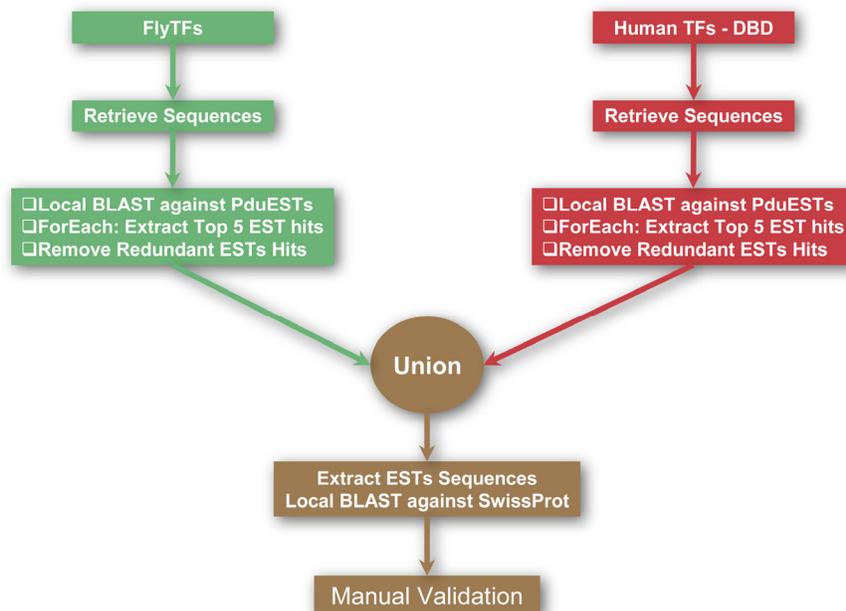


Figure 51. The pipeline used for screening the Transcription Factors represented in *Platynereis* EST sequences.

The figure summarizes the pipeline used to screen for transcription factor genes in *Platynereis* EST sequences. See text for the details.

I designed a relatively simple pipeline (Figure 51) to screen for the transcription factor genes represented in our EST sequences library. I aimed to start from two separate directions, namely from *Drosophila* side and from Human side. Firstly, I retrieved the sequences for all the transcription factors known in *Drosophila* (FlyTFs) (Adryan and Teichmann, 2006) and Human (DBD TFs collection)(Wilson et al., 2008) separately. In the next step I established a local database of all the known EST sequences in *Platynereis*. Subsequently, I used local BLAST program to search for the sequences in *Platynereis* ESTs collection which were similar to the transcription factors' sequences. I performed this search separately for sequences from *Drosophila* side and Human side. Further, I extracted the first five best matched EST sequences from the EST database. Subsequently, I used a filter to remove all the redundant sequences from the collection. Afterwards, the resulting matching ESTs sequences from *Drosophila* side and human side were combined together. In the next step, I did a reverse BLAST search of all the short-listed EST sequences against a local Swissprot database (2008). The best hits were manually analyzed to identify the hits that were matching transcription factors. As a result, I got 118 new candidate transcription factors in our EST collection. The unique ids of all the sequences are listed in Appendix.

4.2.4 mfpBLAST and BrainExplorer development

For searching the WMISEP expression profile database, I developed two search interfaces, namely mfpBLAST and BrainExplorer. I used Matlab programming language for mfpBLAST and Java for BrainExplorer. BrainExplorer was integrated as a plugin in the open source image processing software ImageJ. These programs load the local database of cellular expression profiles on startup and allows sophisticated queries (see Section 2.1.1.8). The source code of both the programs is given in the Appendix.

4.2.5 Image processing algorithm development and testing

For testing the various Image Registration algorithms I used many different systems (see Result section for details). Most of the programs were written in Perl, C++ or Matlab (Source code given in Appendix). Further, I developed several programs for modeling the larval brain of 48 hour old *Platynereis* larvae. The algorithm used is presented in the Result section. For assigning the gene expression patterns to the cellular brain model of *Platynereis* 48 hour old larvae, I developed programs in Matlab using image processing library DIPlib version 1.6 (section 2.1.1.7, Appendix). Briefly, the average intensities for a gene were superimposed on to the model and for each cell in the model, mean intensity value was calculated, which were then converted into Z-scores. A threshold for Z-score was estimated using isothreshold method. The assigned cellular profiles were manually validated and were then stored into a flat file database table.

4.2.6 Clustering analysis

For the clustering analysis I used two different approaches - Heirarchical and Data partitioning. The Heirarchical clustering was performed using Cluster program (Eisen et al., 1998) and pvclust package (Suzuki and Shimodaira, 2006). In all the clustering analysis, uncentered correlation was used as the distance measure and complete linkage was used for the inter-cluster distance measures. Data partitioning based clustering analysis was carried out using hopach package (J. van der Laan and Pollard, 2003). The statistical significance for the clusters of cells was calculated using hopach and for the gene clusters using pvclust. All the resulting trees were visualized using Treeview (Eisen et al., 1998) or MappleTree (<http://mapletree.sourceforge.net>). The cluster images were further processed using Adobe Illustrator.

5. REFERENCES

- (2008). The universal protein resource (UniProt). *Nucleic Acids Res* **36**, D190-5.
- Ackermann, C., Dorresteijn, A., and Fischer, A. (2005). Clonal domains in postlarval *Platynereis dumerilii* (Annelida: Polychaeta). *J Morphol* **266**, 258-80.
- Adoutte, A., Balavoine, G., Lartillot, N., Lespinet, O., Prud'homme, B., and de Rosa, R. (2000). The new animal phylogeny: reliability and implications. *Proc Natl Acad Sci U S A* **97**, 4453-6.
- Adryan, B., and Teichmann, S. A. (2006). FlyTF: a systematic review of site-specific transcription factors in the fruit fly *Drosophila melanogaster*. *Bioinformatics* **22**, 1532-3.
- Alten, H. v. (1910). Zur Phylogenie des Hymenopterengehirns. *Jena Z. Naturwiss.* **46**, 511-590.
- Altmann, C. R., and Brivanlou, A. H. (2001). Neural patterning in the vertebrate embryo. *Int Rev Cytol* **203**, 447-82.
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol* **215**, 403-10.
- Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25**, 3389-402.
- Ando, H., Kobayashi, M., Tsubokawa, T., Uyemura, K., Furuta, T., and Okamoto, H. (2005). Lhx2 mediates the activity of Six3 in zebrafish forebrain growth. *Dev Biol* **287**, 456-68.
- Arendt, D. (2008). The evolution of cell types in animals: emerging principles from molecular studies. *Nat Rev Genet* **9**, 868-82.
- Arendt, D., Denes, A. S., Jekely, G., and Tessmar-Raible, K. (2008). The evolution of nervous system centralization. *Philos Trans R Soc Lond B Biol Sci* **363**, 1523-8.
- Arendt, D., Technau, U., and Wittbrodt, J. (2001). Evolution of the bilaterian larval foregut. *Nature* **409**, 81-5.
- Arendt, D., Tessmar-Raible, K., Snyman, H., Dorresteijn, A. W., and Wittbrodt, J. (2004). Ciliary photoreceptors with a vertebrate-type opsin in an invertebrate brain. *Science* **306**, 869-71.
- Arendt, D., Tessmar, K., de Campos-Baptista, M. I., Dorresteijn, A., and Wittbrodt, J. (2002). Development of pigment-cup eyes in the polychaete *Platynereis dumerilii* and evolutionary conservation of larval eyes in Bilateria. *Development* **129**, 1143-54.
- Benito-Gutierrez, E. (2006). A gene catalogue of the amphioxus nervous system. *Int J Biol Sci* **2**, 149-60.
- Brown, S. M., and Strausfeld, N. J. (2006). Development-dependent and -independent ubiquitin expression in divisions of the cockroach mushroom body. *J Comp Neurol* **496**, 556-71.
- Bulchand, S., Grove, E. A., Porter, F. D., and Tole, S. (2001). LIM-homeodomain gene Lhx2 regulates the formation of the cortical hem. *Mech Dev* **100**, 165-75.
- Carroll, S. B. (2000). Endless forms: the evolution of gene regulation and morphological diversity. *Cell* **101**, 577-80.

- Cayre, M., Buckingham, S. D., Yagodin, S., and Sattelle, D. B. (1999). Cultured insect mushroom body neurons express functional receptors for acetylcholine, GABA, glutamate, octopamine, and dopamine. *J Neurophysiol* **81**, 1-14.
- Chiang, C., Litingtung, Y., Lee, E., Young, K. E., Corden, J. L., Westphal, H., and Beachy, P. A. (1996). Cyclopia and defective axial patterning in mice lacking Sonic hedgehog gene function. *Nature* **383**, 407-13.
- Crittenden, J. R., Skoulakis, E. M., Han, K. A., Kalderon, D., and Davis, R. L. (1998). Tripartite mushroom body architecture revealed by antigenic markers. *Learn Mem* **5**, 38-51.
- D'Haeseleer, P. (2005). How does gene expression clustering work? *Nat Biotech* **23**, 1499.
- Davidson, E. H., and Erwin, D. H. (2006). Gene regulatory networks and the evolution of animal body plans. *Science* **311**, 796-800.
- de Belle, J. S., and Heisenberg, M. (1994). Associative odor learning in *Drosophila* abolished by chemical ablation of mushroom bodies. *Science* **263**, 692-5.
- de Souza, F. S., and Niehrs, C. (2000). Anterior endoderm and head induction in early vertebrate embryos. *Cell Tissue Res* **300**, 207-17.
- Denes, A. S., Jekely, G., Steinmetz, P. R., Raible, F., Snyman, H., Prud'homme, B., Ferrier, D. E., Balavoine, G., and Arendt, D. (2007). Molecular architecture of annelid nerve cord supports common origin of nervous system centralization in bilateria. *Cell* **129**, 277-88.
- Dujardin, F. (1850). Me'moire sur le syste'me nerveux des insectes. *Ann. Sci. Nat. Zool.* **14**, 195-206.
- Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A* **95**, 14863-8.
- Evans, S. M. (1966a). Non-associative avoidance learning in nereid polychaetes. *Anim Behav* **14**, 102-6.
- Evans, S. M. (1966b). Non-associative behavioural modifications in nereid polychaetes. *Nature* **211**, 945-8.
- Evans, S. M., Downie, P.J. (1986). Decision-making processes in the polychaete *Platynereis dumerilii*. *Animal behaviour* **34**, 472-479.
- Farris, S. M. (2005). Evolution of insect mushroom bodies: old clues, new insights. *Arthropod Structure & Development* **34**, 211-234.
- Farris, S. M. (2008). Structural, functional and developmental convergence of the insect mushroom bodies with higher brain centers of vertebrates. *Brain Behav Evol* **72**, 1-15.
- Farris, S. M., and Roberts, N. S. (2005). Coevolution of generalist feeding ecologies and gyrencephalic mushroom bodies in insects. *Proc Natl Acad Sci U S A* **102**, 17394-9.
- Fischer, A., and Dorresteyn, A. (2004). The polychaete *Platynereis dumerilii* (Annelida): a laboratory animal with spiralian cleavage, lifelong segment proliferation and a mixed benthic/pelagic life cycle. *Bioessays* **26**, 314-25.
- Flögel, J. H. L. (1876). Über den feineren Bau des Arthropodengehirns. **49**, 115-120.
- Foley, A. C., and Stern, C. D. (2001). Evolution of vertebrate forebrain development: how many different mechanisms? *J Anat* **199**, 35-52.

- Forel, A. H. (1874). Les fourmis de la Suisse: Systeme´matique, notices anatomiques et physiologiques, architecture, distribution ge´ographique, nouvelles expe´riences et observations de moeurs. *Neue Denkschr. Allg. Schweiz. Gesellschaft Gesammten Naturwiss.* **26**, 1-452.
- Grillenzoni, N., Flandre, A., Lasbleiz, C., and Dura, J. M. (2007). Respective roles of the DRL receptor and its ligand WNT5 in *Drosophila* mushroom body development. *Development* **134**, 3089-97.
- Grove, E. A., Tole, S., Limon, J., Yip, L., and Ragsdale, C. W. (1998). The hem of the embryonic cerebral cortex is defined by the expression of multiple Wnt genes and is compromised in Gli3-deficient mice. *Development* **125**, 2315-25.
- Han, K. A., Millar, N. S., and Davis, R. L. (1998). A novel octopamine receptor with preferential expression in *Drosophila* mushroom bodies. *J Neurosci* **18**, 3650-8.
- Han, P. L., Levin, L. R., Reed, R. R., and Davis, R. L. (1992). Preferential expression of the *Drosophila* rutabaga gene in mushroom bodies, neural centers for learning in insects. *Neuron* **9**, 619-27.
- Hanström, B. (1928). "Vergleichende Anatomie des Nervensystems der Wirbellosen Tiere." Springer, Berlin, Germany,
- Hartenstein, V. (2002). "Drosophila Eye Development." Springer,
- Hebert, J. M., and Fishell, G. (2008). The genetics of early telencephalon patterning: some assembly required. *Nat Rev Neurosci.*
- Heisenberg, M., Borst, A., Wagner, S., and Byers, D. (1985). *Drosophila* mushroom body mutants are deficient in olfactory learning. *J Neurogenet* **2**, 1-30.
- Herzig, M. C., Thor, S., Thomas, J. B., Reichert, H., and Hirth, F. (2001). Expression and function of the LIM homeodomain protein Apterous during embryonic brain development of *Drosophila*. *Dev Genes Evol* **211**, 545-54.
- Heuer, C. M., and Loesel, R. (2008). Immunofluorescence analysis of the internal brain anatomy of *Nereis diversicolor* (Polychaeta, Annelida). *Cell Tissue Res* **331**, 713-24.
- Holden, M., Hill, D. L., Denton, E. R., Jarosz, J. M., Cox, T. C., Rohlfing, T., Goodey, J., and Hawkes, D. J. (2000). Voxel similarity measures for 3-D serial MR brain image registration. *IEEE Trans Med Imaging* **19**, 94-102.
- Holmgren, N. (1916). Zur vergleichenden Anatomie des Gehirns von Polychaeten, Onychophoren, Xiphosuren, Arachniden, Crustaceen, Myriapoden, und Insekten. *K. Svenska Vetensk. Akad. Handl.* **56**, 1-200.
- Ibanez, S., Ng, Cates. (2005). "The ITK Software Guide." Kitware Inc.,
- Ito, K., and Awasaki, T. (2008). Clonal unit architecture of the adult fly brain. *Adv Exp Med Biol* **628**, 137-58.
- Ito, K., and Hotta, Y. (1992). Proliferation pattern of postembryonic neuroblasts in the brain of *Drosophila melanogaster*. *Dev Biol* **149**, 134-48.
- J. van der Laan, M., and Pollard, K. S. (2003). A new algorithm for hybrid hierarchical clustering with visualization and the bootstrap. *Journal of Statistical Planning and Inference* **117**, 275.
- Jacob, F. (1977). Evolution and tinkering. *Science* **196**, 1161-6.
- Jawlowski, H. (1958). Nerve tracks in bee (*Apis mellifera*) running from the sight and antennal organs to the brain. *Ann. Univ. Marie Curie-Sklodowska* **12**, 307-323.

- Jawłowski, H. (1960). On the brain structure of the Symphyta (Hymenoptera). *Bull. Acad. Polon. Sci.* **8**, 265-268.
- Jefferis, G. S., Potter, C. J., Chan, A. M., Marin, E. C., Rohlfsing, T., Maurer, C. R., Jr., and Luo, L. (2007). Comprehensive maps of *Drosophila* higher olfactory centers: spatially segregated fruit and pheromone representation. *Cell* **128**, 1187-203.
- Jekely, G., and Arendt, D. (2007). Cellular resolution expression profiling using confocal detection of NBT/BCIP precipitate by reflection microscopy. *Biotechniques* **42**, 751-5.
- Johard, H. A., Enell, L. E., Gustafsson, E., Trifilieff, P., Veenstra, J. A., and Nassel, D. R. (2008). Intrinsic neurons of *Drosophila* mushroom bodies express short neuropeptide F: relations to extrinsic neurons expressing different neurotransmitters. *J Comp Neurol* **507**, 1479-96.
- Joiner, W. J., Crocker, A., White, B. H., and Sehgal, A. (2006). Sleep in *Drosophila* is regulated by adult mushroom bodies. *Nature* **441**, 757-60.
- Jonescu, C. N. (1909). Vergleichende Untersuchungen u̇ber das Gehirn der Honigbiene. *Jena Z. Naturwiss.* **45**, 111-180.
- Kandel, E., and Abel, T. (1995). Neuropeptides, adenylyl cyclase, and memory storage. *Science* **268**, 825-6.
- Kerner, P., Zelada Gonzalez, F., Le Gouar, M., Ledent, V., Arendt, D., and Vervoort, M. (2006). The expression of a hunchback ortholog in the polychaete annelid *Platynereis dumerilii* suggests an ancestral role in mesoderm development and neurogenesis. *Dev Genes Evol* **216**, 821-8.
- King, M. C., and Wilson, A. C. (1975). Evolution at two levels in humans and chimpanzees. *Science* **188**, 107-16.
- Klein, S., Staring, M., and Pluim, J. P. (2007). Evaluation of optimization methods for nonrigid medical image registration using mutual information and B-splines. *IEEE Trans Image Process* **16**, 2879-90.
- Kobayashi, M., Michaut, L., Ino, A., Honjo, K., Nakajima, T., Maruyama, Y., Mochizuki, H., Ando, M., Ghangrekar, I., Takahashi, K., Saigo, K., Ueda, R., Gehring, W. J., and Furukubo-Tokunaga, K. (2006). Differential microarray analysis of *Drosophila* mushroom body transcripts using chemical ablation. *Proc Natl Acad Sci U S A* **103**, 14417-22.
- Koh, K., Evans, J. M., Hendricks, J. C., and Sehgal, A. (2006). A *Drosophila* model for age-associated changes in sleep:wake cycles. *Proc Natl Acad Sci U S A* **103**, 13843-7.
- Kurusu, M., Nagao, T., Walldorf, U., Flister, S., Gehring, W. J., and Furukubo-Tokunaga, K. (2000). Genetic control of development of the mushroom bodies, the associative learning centers in the *Drosophila* brain, by the *eyeless*, *twins* of *eyeless*, and *Dachshund* genes. *Proc Natl Acad Sci U S A* **97**, 2140-4.
- Lisa Gottesfeld, B. (1992). A survey of image registration techniques. *ACM Comput. Surv.* **24**, 325-376.
- Lu, C. C., Brennan, J., and Robertson, E. J. (2001). From fertilization to gastrulation: axis formation in the mouse embryo. *Curr Opin Genet Dev* **11**, 384-92.
- Mangale, V. S., Hirokawa, K. E., Satyaki, P. R., Gokulchandran, N., Chikbire, S., Subramanian, L., Shetty, A. S., Martynoga, B., Paul, J., Mai, M. V., Li, Y., Flanagan, L. A., Tole, S., and Monuki, E. S. (2008). *Lhx2* selector activity

- specifies cortical identity and suppresses hippocampal organizer fate. *Science* **319**, 304-9.
- McGinnis, W., Garber, R. L., Wirz, J., Kuroiwa, A., and Gehring, W. J. (1984). A homologous protein-coding sequence in *Drosophila* homeotic genes and its conservation in other metazoans. *Cell* **37**, 403-8.
- Mizunami, M., Iwasaki, M., Okada, R., and Nishikawa, M. (1998a). Topography of four classes of Kenyon cells in the mushroom bodies of the cockroach. *J Comp Neurol* **399**, 162-75.
- Mizunami, M., Weibrecht, J. M., and Strausfeld, N. J. (1998b). Mushroom bodies of the cockroach: their participation in place memory. *J Comp Neurol* **402**, 520-37.
- Monuki, E. S., Porter, F. D., and Walsh, C. A. (2001). Patterning of the dorsal telencephalon and cerebral cortex by a roof plate-Lhx2 pathway. *Neuron* **32**, 591-604.
- Moon, R. T., and Kimelman, D. (1998). From cortical rotation to organizer gene expression: toward a molecular explanation of axis specification in *Xenopus*. *Bioessays* **20**, 536-45.
- Muller, W. A. (1973). [Autoradiographic studies on the synthetic activity of neurosecretory cells in the brain of *Platynereis dumerilii* during sexual development and regeneration]. *Z Zellforsch Mikrosk Anat* **139**, 487-510.
- Murakami, Y., Uchida, K., Rijli, F. M., and Kuratani, S. (2005). Evolution of the brain developmental plan: Insights from agnathans. *Dev Biol* **280**, 249-59.
- Nomura, T., Takahashi, M., Hara, Y., and Osumi, N. (2008). Patterns of neurogenesis and amplitude of Reelin expression are essential for making a mammalian-type cortex. *PLoS ONE* **3**, e1454.
- Noveen, A., Daniel, A., and Hartenstein, V. (2000). Early development of the *Drosophila* mushroom body: the roles of *eyeless* and *dachshund*. *Development* **127**, 3475-88.
- Orrhage, L. (1993). On the microanatomy of the cephalic nervous system of Nereidae (Polychaeta), with a preliminary discussion of some earlier theories on the segmentation of the polychaete brain. *Acta zoologica* **74**, 145-172.
- Pitman, J. L., McGill, J. J., Keegan, K. P., and Allada, R. (2006). A dynamic role for the mushroom bodies in promoting sleep in *Drosophila*. *Nature* **441**, 753-6.
- Porter, F. D., Drago, J., Xu, Y., Cheema, S. S., Wassif, C., Huang, S. P., Lee, E., Grinberg, A., Massalas, J. S., Bodine, D., Alt, F., and Westphal, H. (1997). Lhx2, a LIM homeobox gene, is required for eye, forebrain, and definitive erythrocyte development. *Development* **124**, 2935-44.
- Prokop, A., and Technau, G. M. (1991). The origin of postembryonic neuroblasts in the ventral nerve cord of *Drosophila melanogaster*. *Development* **111**, 79-88.
- Prud'homme, B., de Rosa, R., Arendt, D., Julien, J. F., Pajaziti, R., Dorresteyn, A. W., Adoutte, A., Wittbrodt, J., and Balavoine, G. (2003). Arthropod-like expression patterns of engrailed and wingless in the annelid *Platynereis dumerilii* suggest a role in segment formation. *Curr Biol* **13**, 1876-81.
- Raff, R. A. (2000). Evo-devo: the evolution of a new discipline. *Nat Rev Genet* **1**, 74-9.
- Raible, F., and Arendt, D. (2004). Metazoan evolution: some animals are more equal than others. *Curr Biol* **14**, R106-8.
- Raible, F., Tessmar-Raible, K., Osoegawa, K., Wincker, P., Jubin, C., Balavoine, G., Ferrier, D., Benes, V., de Jong, P., Weissenbach, J., Bork, P., and Arendt, D.

- (2005). Vertebrate-type intron-rich genes in the marine annelid *Platynereis dumerilii*. *Science* **310**, 1325-6.
- Rallu, M., Corbin, J. G., and Fishell, G. (2002). Parsing the prosencephalon. *Nat Rev Neurosci* **3**, 943-51.
- Rohlfing, T., and Maurer, C. R., Jr. (2003). Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees. *IEEE Trans Inf Technol Biomed* **7**, 16-25.
- Rohlfing, T., Schaupp, F., Haddad, D., Brandt, R., Haase, A., Menzel, R., and Maurer, C. R., Jr. (2005). Unwarping confocal microscopy images of bee brains by nonrigid registration to a magnetic resonance microscopy image. *J Biomed Opt* **10**, 024018.
- Rokas, A., and Carroll, S. B. (2006). Bushes in the tree of life. *PLoS Biol* **4**, e352.
- Sasai, Y., and De Robertis, E. M. (1997). Ectodermal patterning in vertebrate embryos. *Dev Biol* **182**, 5-20.
- Schier, A. F. (2001). Axis formation and patterning in zebrafish. *Curr Opin Genet Dev* **11**, 393-404.
- Shimamura, K., and Rubenstein, J. L. (1997). Inductive interactions direct early regionalization of the mouse forebrain. *Development* **124**, 2709-18.
- Staring, M., Klein, S., and Pluim, J. P. (2007). A rigidity penalty term for nonrigid registration. *Med Phys* **34**, 4098-108.
- Strausfeld, N. J., Hansen, L., Li, Y., Gomez, R. S., and Ito, K. (1998). Evolution, discovery, and interpretations of arthropod mushroom bodies. *Learn Mem* **5**, 11-37.
- Strausfeld, N. J., Strausfeld, C. M., Stowe, S., Rowell, D., and Loesel, R. (2006). The organization and evolutionary implications of neuropils and their neurons in the brain of the onychophoran *Euperipatoides rowelli*. *Arthropod Struct Dev* **35**, 169-96.
- Suzuki, R., and Shimodaira, H. (2006). Pvcust: an R package for assessing the uncertainty in hierarchical clustering. *Bioinformatics* **22**, 1540-2.
- Tao, W., and Lai, E. (1992). Telencephalon-restricted expression of BF-1, a new member of the HNF-3/fork head gene family, in the developing rat brain. *Neuron* **8**, 957-66.
- Tessmar-Raible, K., and Arendt, D. (2003). Emerging systems: between vertebrates and arthropods, the Lophotrochozoa. *Curr Opin Genet Dev* **13**, 331-40.
- Tessmar-Raible, K., Raible, F., Christodoulou, F., Guy, K., Rembold, M., Hausen, H., and Arendt, D. (2007). Conserved sensory-neurosecretory cell types in annelid and fish forebrain: insights into hypothalamus evolution. *Cell* **129**, 1389-400.
- Tessmar-Raible, K., Steinmetz, P. R., Snyman, H., Hassel, M., and Arendt, D. (2005). Fluorescent two-color whole mount in situ hybridization in *Platynereis dumerilii* (Polychaeta, Annelida), an emerging marine molecular model for evolution and development. *Biotechniques* **39**, 460, 462, 464.
- Theil, T., Alvarez-Bolado, G., Walter, A., and Ruther, U. (1999). Gli3 is required for Emx gene expression during dorsal telencephalon development. *Development* **126**, 3561-71.

- Thompson, J. D., Gibson, T. J., Plewniak, F., Jeanmougin, F., and Higgins, D. G. (1997). The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic Acids Res* **25**, 4876-82.
- Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res* **22**, 4673-80.
- Tole, S., Ragsdale, C. W., and Grove, E. A. (2000). Dorsoventral patterning of the telencephalon is disrupted in the mouse mutant extra-toes(J). *Dev Biol* **217**, 254-65.
- Truman, J. W., and Bate, M. (1988). Spatial and temporal patterns of neurogenesis in the central nervous system of *Drosophila melanogaster*. *Dev Biol* **125**, 145-57.
- Urbach, R., and Technau, G. M. (2003). Molecular markers for identified neuroblasts in the developing brain of *Drosophila*. *Development* **130**, 3621-37.
- Vowles, D. M. (1964). Olfactory Learning And Brain Lesions In The Wood Ant (*Formica Rufa*). *J Comp Physiol Psychol* **58**, 105-11.
- Wilson, D., Charoensawan, V., Kummerfeld, S. K., and Teichmann, S. A. (2008). DBD--taxonomically broad transcription factor predictions: new content and functionality. *Nucleic Acids Res* **36**, D88-92.
- Wonders, C. P., and Anderson, S. A. (2006). The origin and specification of cortical interneurons. *Nat Rev Neurosci* **7**, 687-96.
- Woods, R. P., Cherry, S. R., and Mazziotta, J. C. (1992). Rapid automated algorithm for aligning and reslicing PET images. *J Comput Assist Tomogr* **16**, 620-33.
- Woods, R. P., Grafton, S. T., Holmes, C. J., Cherry, S. R., and Mazziotta, J. C. (1998). Automated image registration: I. General methods and intrasubject, intramodality validation. *J Comput Assist Tomogr* **22**, 139-52.
- Woods, R. P., Mazziotta, J. C., and Cherry, S. R. (1993). MRI-PET registration with automated algorithm. *J Comput Assist Tomogr* **17**, 536-46.
- Yoo, T. S., Ackerman, M. J., Lorensen, W. E., Schroeder, W., Chalana, V., Aylward, S., Metaxas, D., and Whitaker, R. (2002). Engineering and algorithm design for an image processing Api: a technical report on ITK--the Insight Toolkit. *Stud Health Technol Inform* **85**, 586-92.
- Yuan, Q., Joiner, W. J., and Sehgal, A. (2006). A sleep-promoting role for the *Drosophila* serotonin receptor 1A. *Curr Biol* **16**, 1051-62.

6 APPENDIX

6.1 Source Codes of Programs

6.1.1 Image registration algorithms testing and development

wmisp-affine.cxx

This program does affine (rigid body) registration of two 3D images using mutual information as the similarity metric, 3rd order B-Spline as the interpolator, 5 levels of blurring and the regular gradient descent optimizer. It requires ITK libraries (open source library of image processing algorithms released under BSD license approved by the Open Source Initiative, <http://www.itk.org/HTML/Copyright.htm>) and cmake (cross-platform make, <http://www.cmake.org>) for platform-independent compilation.

```

#if defined( _MSC_VER)
#pragma warning (disable : 4786)
#endif

// Includes Start

#include "itkMultiResolutionImageRegistrationMethod.h"
#include "itkImage.h"
#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"
#include "itkAffineTransform.h"
#include "itkMattesMutualInformationImageToImageMetric.h"
#include "itkRegularStepGradientDescentOptimizer.h"
#include "itkRecursiveMultiResolutionPyramidImageFilter.h"
#include "itkResampleImageFilter.h"
#include "itkCastImageFilter.h"
#include "itkCommand.h"
#include "itkBSplineInterpolateImageFunction.h"

// Includes End

// Observer Class

class CommandIterationUpdate : public itk::Command
{
public:
    typedef CommandIterationUpdate      Self;
    typedef itk::Command                Superclass;
    typedef itk::SmartPointer<Self>    Pointer;
    itkNewMacro(Self) ;
protected:
    CommandIterationUpdate(): m_CumulativeIterationIndex(0) {};

public:
    typedef itk::RegularStepGradientDescentOptimizer OptimizerType;
    typedef const OptimizerType * OptimizerPointer;

    void Execute(itk::Object *caller, const itk::EventObject & event) {
        Execute((const itk::Object *)caller, event);
    }
}

```

```

void Execute(const itk::Object *object, const itk::EventObject & event){
    OptimizerPointer optimizer = dynamic_cast<OptimizerPointer>(object) ;
    if (!(itk::IterationEvent().CheckEvent(&event))){
        return;
    }
    std::cout << "Entering New Resolution Level" << std::endl ;
    std::cout << optimizer->GetCurrentIteration() << " ";
    std::cout << optimizer->GetValue() << " ";
    std::cout << optimizer->GetCurrentPosition() << " ";
    std::cout << m_CumulativeIterationIndex++ << std::endl;
}

private:
unsigned int m_CumulativeIterationIndex;
};

// Command Observer to modify parameters at each resolution level

template <typename TRegistration>
class RegistrationInterfaceCommand : public itk::Command {

public:
    typedef RegistrationInterfaceCommand Self ;
    typedef itk::Command Superclass;
    typedef itk::SmartPointer<Self> Pointer;
    itkNewMacro(Self);

protected:
    RegistrationInterfaceCommand() {};

public:
    typedef TRegistration RegistrationType;
    typedef RegistrationType * RegistrationPointer;
    typedef itk::RegularStepGradientDescentOptimizer OptimizerType ;
    typedef OptimizerType * OptimizerPointer;

    void Execute(itk::Object * object, const itk::EventObject & event){
        if (!(itk::IterationEvent().CheckEvent(&event))){
            return;
        }
        RegistrationPointer registration = dynamic_cast<RegistrationPointer>(
object );
        OptimizerPointer optimizer = dynamic_cast< OptimizerPointer
>(registration->GetOptimizer() );
        std::cout << "Level Number: " << registration->GetCurrentLevel() <<
std::endl;
        if ( registration->GetCurrentLevel() == 0 ){
            optimizer->SetMaximumStepLength( 16.00 );
            optimizer->SetMinimumStepLength( 1.0 );
            optimizer->SetGradientMagnitudeTolerance(1.0 / 1000000000);
        }
        else {
            optimizer->SetMaximumStepLength(optimizer->GetMaximumStepLength() /
2.0 );
            optimizer->SetMinimumStepLength(optimizer->GetMinimumStepLength() /
2.0 );
            optimizer->SetGradientMagnitudeTolerance(optimizer-
>GetGradientMagnitudeTolerance() / 10) ;
        }

        void Execute(const itk::Object * , const itk::EventObject &){
            return;
        }
};

// Main function

```

```

int main(int argc, char *argv[]) {
    if (argc < 4){
        std::cerr << "Missing Parameters " << std::endl;
        std::cerr << "Usage: " << argv[0];
        std::cerr << " fixedImageFile movingImageFile ";
        std::cerr << " outputImagefile" << std::endl;
        return EXIT_FAILURE;
    }

    // Pixels Type and Dimension
    const int Dimension = 3;
    typedef unsigned short PixelType;
    typedef itk::Image<PixelType,Dimension> FixedImageType;
    typedef itk::Image<PixelType,Dimension> MovingImageType;
    typedef float InternalPixelType;
    //

    //Transform
    typedef itk::Image<InternalPixelType,Dimension> InternalImageType;
    typedef itk::AffineTransform<double,Dimension> TransformType;
    //

    //Optimizer
    typedef itk::RegularStepGradientDescentOptimizer OptimizerType;
    typedef OptimizerType::ScalesType OptimizerScalesType;
    //

    //Interpolator
    typedef itk::BSplineInterpolateImageFunction<InternalImageType> InterpolatorType;
    //

    //Metric
    typedef
itk::MattesMutualInformationImageToImageMetric<InternalImageType, InternalImageType>
MetricType ;
    //

    // Pyramids Definition
    typedef
itk::MultiResolutionImageRegistrationMethod<InternalImageType, InternalImageType>
RegistrationType;
    typedef
itk::RecursiveMultiResolutionPyramidImageFilter<InternalImageType, InternalImageType>
FixedImagePyramidType;
    typedef
itk::RecursiveMultiResolutionPyramidImageFilter<InternalImageType, InternalImageType>
MovingImagePyramidType;
    //

    OptimizerType::Pointer optimizer = OptimizerType::New()
;
    InterpolatorType::Pointer interpolator = InterpolatorType::New() ;
    RegistrationType::Pointer registration = RegistrationType::New() ;
    MetricType::Pointer metric = MetricType::New() ;
    TransformType::Pointer transform = TransformType::New() ;

    FixedImagePyramidType::Pointer fixedImagePyramid = FixedImagePyramidType::New()
;
    MovingImagePyramidType::Pointer movingImagePyramid = MovingImagePyramidType::New()
;

    unsigned int startingShrinkFactors[3] = {8,8,4};
    fixedImagePyramid->SetStartingShrinkFactors(startingShrinkFactors);
    movingImagePyramid->SetStartingShrinkFactors(startingShrinkFactors);

    const unsigned int bsplineOrder = 3;
    interpolator->SetSplineOrder(bsplineOrder) ;
    registration->SetOptimizer(optimizer) ;

```

```

registration->SetInterpolator(interpolator) ;
registration->SetMetric(metric) ;
registration->SetTransform(transform) ;
registration->SetFixedImagePyramid( fixedImagePyramid );
registration->SetMovingImagePyramid( movingImagePyramid );

typedef itk::ImageFileReader<FixedImageType>
FixedImageReaderType;
typedef itk::ImageFileReader<MovingImageType>      MovingImageReaderType;

FixedImageReaderType::Pointer fixedImageReader = FixedImageReaderType::New() ;
MovingImageReaderType::Pointer movingImageReader = MovingImageReaderType::New() ;

fixedImageReader->SetFileName(argv[1]) ;
movingImageReader->SetFileName(argv[2]) ;

typedef itk::CastImageFilter<FixedImageType,InternalImageType>
FixedCastFilterType;
typedef itk::CastImageFilter<MovingImageType,InternalImageType>
MovingCastFilterType;

FixedCastFilterType::Pointer fixedCaster = FixedCastFilterType::New() ;
MovingCastFilterType::Pointer movingCaster = MovingCastFilterType::New() ;

fixedCaster->SetInput(fixedImageReader->GetOutput()) ;
movingCaster->SetInput(movingImageReader->GetOutput()) ;

registration->SetFixedImage(fixedCaster->GetOutput());
registration->SetMovingImage(movingCaster->GetOutput());

fixedCaster->Update() ;
registration->SetFixedImageRegion(fixedCaster->GetOutput()-
>GetLargestPossibleRegion());

//Initialize transform parameters by using identity transform

transform->SetIdentity() ;
registration->SetInitialTransformParameters(transform->GetParameters()) ;

OptimizerScalesType optimizerScales( transform->GetNumberOfParameters() );
optimizerScales[0] = 1.0; // scale for M11
optimizerScales[1] = 1.0; // scale for M12
optimizerScales[2] = 1.0; // scale for M13
optimizerScales[3] = 1.0; // scale for M21
optimizerScales[4] = 1.0; // scale for M22
optimizerScales[5] = 1.0; // scale for M23
optimizerScales[6] = 1.0; // scale for M31
optimizerScales[7] = 1.0; // scale for M32
optimizerScales[8] = 1.0; // scale for M33

optimizerScales[9] = 1.0 / 50000.0; // scale for translation on X
optimizerScales[10] = 1.0 / 50000.0; // scale for translation on y
optimizerScales[11] = 1.0 / 50000.0; // scale for translation on z

optimizer->SetScales(optimizerScales);

metric->SetNumberOfHistogramBins(32);
metric->SetNumberOfSpatialSamples(100000) ;

metric->ReinitializeSeed(76926294) ;

optimizer->SetNumberOfIterations(500) ;
optimizer->SetGradientMagnitudeTolerance(1.0 / 10000000);

// Command Observer

CommandIterationUpdate::Pointer observer = CommandIterationUpdate::New() ;
optimizer->AddObserver(itk::IterationEvent(), observer) ;
typedef RegistrationInterfaceCommand<RegistrationType> CommandType ;
CommandType::Pointer command = CommandType::New();

```

```

registration->AddObserver( itk::IterationEvent(), command );
registration->SetNumberOfLevels(5) ;

// Registration Process begins
try {
    std::cout << "Attempting to start Registration ..." << std::endl;
    registration->StartRegistration();
}
catch(itk::ExceptionObject & err) {
    std::cout << "ExceptionObject caught !" << std::endl;
    std::cout << err << std::endl;
    return EXIT_FAILURE;
}

typedef RegistrationType::ParametersType ParametersType;
ParametersType finalParameters = registration->GetLastTransformParameters();
unsigned int numberOfIterations = optimizer->GetCurrentIteration();

double bestValue = optimizer->GetValue();

std::cout << " Iterations    = " << numberOfIterations << std::endl;
std::cout << " Metric value  = " << bestValue          << std::endl;

typedef itk::ResampleImageFilter<MovingImageType,FixedImageType>
ResampleFilterType;
typedef itk::BSplineInterpolateImageFunction<InternalImageType>
ResampleInterpolatorType;

ResampleInterpolatorType::Pointer resampleInterpolator =
ResampleInterpolatorType::New();
const unsigned int resampleInterpolatorOrder = 3;
resampleInterpolator->SetSplineOrder(resampleInterpolatorOrder) ;

TransformType::Pointer finalTransform = TransformType::New();
finalTransform->SetParameters( finalParameters );
ResampleFilterType::Pointer resample = ResampleFilterType::New();
resample->SetTransform( finalTransform );
resample->SetInput( movingImageReader->GetOutput() );
FixedImageType::Pointer fixedImage = fixedImageReader->GetOutput();
PixelType backgroundGrayLevel = 0;

resample->SetSize( fixedImage->GetLargestPossibleRegion().GetSize() );
resample->SetOutputOrigin( fixedImage->GetOrigin() );
resample->SetOutputSpacing( fixedImage->GetSpacing() );
resample->SetDefaultPixelValue( backgroundGrayLevel );

typedef unsigned char OutputPixelType;
typedef itk::Image< OutputPixelType, Dimension > OutputImageType;
typedef itk::CastImageFilter<FixedImageType,OutputImageType> CastFilterType;
typedef itk::ImageFileWriter<OutputImageType> WriterType;

WriterType::Pointer writer = WriterType::New();
CastFilterType::Pointer caster = CastFilterType::New();

writer->SetFileName( argv[3] );
caster->SetInput( resample->GetOutput() );
writer->SetInput( caster->GetOutput() );
writer->Update();

return EXIT_SUCCESS;
}

```

6.1.2 Averaging expression patterns

The program is written in MATLAB and requires DIPlib (www.diplib.org) for functionality. Briefly, the program reads all the expression images of a gene, normalize the individual images by transforming intensity in a way to make mean equals to 0 and standard deviation equals to 1. Further, the normalized images are averaged. In the last step, the average image is mapped back to [0,255] range. Note: The final average image may look brighter than raw input images, due to the final normalization step.

doavg_normalized.m

```
function im_avg = doavg_normalized(gene_dir,avg_name)
% Program syntax
% im_avg = doavg_normalized(gene_dir,avg_name)
% where gen_dir = name of the directory where the images are in Biorad
% format, avg_name = name you would like to append to the file name for the
% average images
%
%Change out_dir to indicate the full path of the directory where you want
% to save the average images
out_dir = 'C:\tomer\ImageRegistration\48hpf\AvgBrainVer5AsRef\ScannedByRaju\Averages\';

% Mask image covering the average brain model
mask = readim('C:\tomer\ImageRegistration\48hpf\Reference_Images\Pdu48EmbryonicShell_1-8Rmd.PIC','PIC');
mask = mask > 0;

% im_dir = name of the directory where the folders for the images are
im_dir = 'C:\tomer\ImageRegistration\48hpf\AvgBrainVer5AsRef\ScannedByRaju\';

data_dir = [im_dir gene_dir '*ch02*'];
im_data_file = dir(data_dir);
sz = size(im_data_file);
no_ims = sz(1);
disp(sprintf('No. of Images %d,no_ims));
im_avg = readim([im_dir gene_dir '\ im_data_file(1).name'],'PIC');
im_avg = dip_image(im_avg,'sfloat');
im_avg(:,:,) = 0;

for n = 1:no_ims
    im = readim([im_dir gene_dir '\ im_data_file(n).name'],'PIC');
    im(~mask) = 0;
    im_mean = mean(im(mask));
    im_var = sqrt(var(im(mask)));
    disp(sprintf('Mean=%5.5g, Sqrt Var=%5.5g',im_data_file(n).name,im_mean,im_var));
    im_norm = (im - im_mean)/im_var;
    im_avg = im_avg + im_norm/no_ims;
end

im_avg(~mask) = 0;
lower_stretch_bound_percentile = 1;
upper_stretch_bound_percentile = 99;
lower_stretch_bound = percentile(im_avg(im_avg>0),lower_stretch_bound_percentile);
upper_stretch_bound = percentile(im_avg(im_avg>0),upper_stretch_bound_percentile);
disp(sprintf('Stretching Average Image with Lower Bound = %1.2g percentile, Upper Bound = %3.4g
percentile',lower_stretch_bound,upper_stretch_bound));
```

```

disp(sprintf('Lower Bound = %3.4g, Upper Bound = %3.4g',lower_stretch_bound,upper_stretch_bound));
im_avg = (im_avg - lower_stretch_bound)*(255 - 0)/(upper_stretch_bound - lower_stretch_bound) + 0;
im_avg(im_avg<0) = 0;
im_avg = dip_image(im_avg,'uint8');
out_fname = [out_dir gene_dir avg_name 'Of' num2str(no_ims) 'images' 'Lo' num2str(lower_stretch_bound_percentile) '_Hi'
num2str(upper_stretch_bound_percentile) 'b'];
writeim(im_avg,out_fname,'ICsv2',0,1);
out_fname = [out_fname '.ics'];
disp(sprintf('%s written!',out_fname));

```

6.1.3 Cellular modelling of *Platynereis* larval brain

The algorithms used for the cellular modeling of *Platynereis* larval brain are discussed in Results section. The code is written in MATLAB and requires DIPlib (www.diplib.org) for functionality.

Command line scripts:

```

fname = 'C:\tomer\NuclearModelling48hpf\Model_100608_2\Data\ver5Avg_DachD1148-1-ch03_warp_m0g40c4e1e-1x16r3.PIC';
ofname = 'C:\tomer\NuclearModelling48hpf\Model_100608_2\Data\ver5Avg_DachD1148-1-ch03_warp_m0g40c4e1e-1x16r3.txt';
aspect = [0.3756 0.3756 1]; % Voxel size of the images
thr = 0;
[dapiSignal,dapiImg,dapiImg_model,nucleiCoords,modelledNuclei,no_nuc] = pdu_find_nuc_model(fname,ofname,aspect,thr);

```

```

% before next step, I used Pointpicker plugin in ImageJ to manually check and correct the nuclei center coordinates
pts_fname = 'C:\tomer\NuclearModelling48hpf\Model_100608_2\Data\ver5Avg_DachD1148-1-ch03_warp_m0g40c4e1e-1x16r3-
corrected-header_rmd.txt';
[modelledNuclei1,no_nuc] = pdu_find_model(pts_fname,dapiImg,dapiSignal,aspect);

```

```

% Calculate the coordinates of the center of all the modeled nuclei
msr1 = measure(modelledNuclei1,[],({'center'}));
pts = msr1.center;
pts1 = round(pts);

```

```

[modelledNuclei_Final,no_nuc] = pdu_find_model_Centers(pts1,dapiImg,dapiSignal,aspect);

```

```

msr = measure(modelledNuclei_Final,[],({'size','center'}));
pts = msr.center;
pts = round(pts);

```

```

% Calculate the coordinates of the center of all the modeled nuclei
cfname = 'C:\tomer\NuclearModelling48hpf\Model_100608_2\Model_100608_final_coords.txt';
[no_nuc] = write_centerCord_in_pointpicker(pts,cfname);

```

```

% Nuclei smaller than 200 voxels were removed and nuclei were modelles again to cover the signals
lower_sz_thresh = 200;
[fnuclei_sizeCut,no_nuc] = pdu_find_model_rm_small_nuc(pts_fname, dapiImg, dapiSignal,aspect,lower_sz_thresh);
writeim(fnuclei_sizeCut,'C:\tomer\NuclearModelling48hpf\Model_100608_2\Model_100608_final_200thresh','ICsv2',0,1);

```

pdu_find_nuc_model.m

```

function [dapiSignal,dapiImg,dapiImg_model,nucleiCoords,modelledNuclei,no_nuc] =
pdu_find_nuc_model(fname,ofname,aspect,thr)
% This function takes DAPI staining (assuming it is processed by imageJ so as
% to clear any signal outside the embryo), do signal attenuation
% correction, find local maxima as nuclei position and model the full
% nuclei. In addition it saves the center of modelled nuclei in Pointpicker format for manual validation.
% fname = Name of file containing the Dapi image stack in Biorad PIC format
% ofname = output file name for saving the coordinates of center of
% modelled nuclei
% aspect = a vector containing dimeninsional information for voxels
% thr = threshold for background

```

% output variables are:

```

if nargin<4
thr = 0;
end
if nargin<3
aspect = [0.3756 0.3756 1];
end
if nargin<2
error('Not enough input arguments')
end
aspectRatio = aspect(1)/aspect(3);

```

% Load Image Data

```

imgdata = readim(fname,'PIC'); % Image should be processed in the ImageJ
mask = readim('C:\tomer\ImageRegistration\48hpf\Reference_Images\Pdu48EmbryonicShell.PIC','PIC');
imgdata(~mask) = 0;
dapiImg = dip_image(squeeze(imgdata),'uint16');
clear imgdata
ms1 = dapiImg > thr ;
dapiImg(~ms1) = 0;
clear thr

```

% Read 3D shell

```

mask = readim('C:\tomer\ImageRegistration\48hpf\Reference_Images\Pdu48EmbryonicShell.PIC','PIC');
shell = mask > 0 ;
clear mask

```

% Attenuation Correction

```

oldGaussianMethod = setgaussmode('fir');
dapiImg_smooth = gaussf(dapiImg,2*[1,1,aspectRatio]);
setgaussmode(oldGaussianMethod);

```

% Local threshold within shell

```

localThreshold = dapiImg_smooth;
localThreshold(~shell) = 0;
localThreshold = gaussf_iir(localThreshold,10*[1,1,aspectRatio]);
aspNormalize = gaussf_iir(+shell,10*[1,1,aspectRatio]);
localThreshold = localThreshold(shell)/aspNormalize(shell);
clear aspNormalize
dapiSignal = newim(shell,'bin');
dapiSignal(shell) = dapiImg_smooth(shell)>localThreshold;
clear lt localThreshold

```

% Sample local average intensity

```

aspTemp = dapiImg_smooth;
aspTemp(~dapiSignal) = 0;
aspTemp = gaussf_iir(aspTemp,10*[1,1,aspectRatio]);
aspNormalize = gaussf_iir(+dapiSignal,10*[1,1,aspectRatio]);
mask = aspNormalize>1e-2;
aspTemp = aspTemp(mask)/aspNormalize(mask);
clear aspNormalize dapiSignal

```

% Attenuation correction

```

dapiImg = dip_image(dapiImg,'sfloat');
dapiImg(mask) = dapiImg(mask) / aspTemp;
dapiImg(~mask) = 0; % Final dapiImg
dapiImg_smooth(mask) = dapiImg_smooth(mask) / aspTemp;
dapiImg_smooth(~mask) = 0;
clear aspTemp mask

```

% Threshold selection within shell

```

[tmp,th] = threshold(dapiImg_smooth(shell),'isodata');
clear tmp
dapiSignal = dapiImg_smooth > th; % Final DNA mask
clear th dapiImg_smooth shell

```

```

%Find minima for nuclei
size_of_parabolic_kernel = 30;
oldGaussianMethod = setgaussmode('fir');
dapiImg_smooth = dapiImg;
dapiImg_smooth = gaussf(dapiImg_smooth,2*[1,1,aspectRatio]);
dapiImg_smooth = -dapiImg_smooth;
setgaussmode(oldGaussianMethod);

% Local maxima
nucleiCoords = dip_localminima(dapiImg_smooth,dapiSignal,3,100,50,0);
clear dapiImg_smooth
nucleiCoords = dip_image(nucleiCoords,'uint16');
msrO = measure(nucleiCoords,[],'center');
no_nuc = size(msrO,1);
id = msrO.id;
no_nuc
cX = msrO.Center(1,:);
cY = msrO.Center(2,:);
cZ = msrO.Center(3,:);
maxid = max(id);
lab = zeros(1,maxid);
lab(id) = id;
clear id maxid msrO
lab = uint16([0,lab]);
nucleiCoords = lut(nucleiCoords,lab);
clear lab

%Write Nuclei in Poinpicker format
[cZ_sort,1] = sort(cZ);
clear cZ_sort
cZ = cZ(1);
cY = cY(1);
cX = cX(1);
cZ = cZ + 1;
cZ = round(cZ);
cY = round(cY);
cX = round(cX);

fid = fopen(ofname,'w');
fprintf(fid,'%s %s %s %s %s\n','point','x','y','slice','color');
prev_id = cZ(1);
count_id = -1 ;

for ii = 1:length(cZ)
    if cZ(ii) == prev_id
        count_id = count_id + 1;
    else
        count_id = 0;
    end
    prev_id = cZ(ii);
    fprintf(fid,'%d %d %d %d %d\n',[count_id cX(ii) cY(ii) cZ(ii) count_id]);
end

fclose(fid);
clear cX cY cZ count_id prev_id

%Model Nuclei
tmp = ~dapiSignal;
dapiImg_model = dapiImg;
dapiImg_model(tmp) = 4095;
offset = erosion(dapiImg_model,size_of_parabolic_kernel*[1,1,aspectRatio],'parabolic');
offset = offset(dapiSignal);
dapiImg_model(tmp) = 0;
clear tmp
range = dilation(dapiImg_model,size_of_parabolic_kernel*[1,1,aspectRatio],'parabolic');
range = max(range(dapiSignal)-offset,1); % Avoid division by zero.
tmp = (dapiImg_model(dapiSignal)-offset)/range;
clear offset range

```

```

tmp = 1-clip(tmp,1,0);
dapiImg_model(dapiSignal) = exp(3*tmp);
clear tmp size_of_parabolic_kernel

% Find the full nuclei
modelledNuclei = nucleiCoords;
modelledNuclei = dip_growregionsweighted(modelledNuclei,dapiImg_model,dapiSignal,aspect,5,[]);
clear oldGaussianMethod aspect

```

pdu_find_model.m

```

function [modelledNuclei,no_nuc] = pdu_find_model(pts_fname,dapiImg,dapiSignal,aspect)
%
% pts_fname = file containing coordinates for nuclei. Make sure there is no
% header left and coordinates are X,Y, and Z format
% dapiImg = Signal attenuation corrected image
% dapiSignal = output from pdu_find_nuc_model.m
% aspect = a vector containing voxel dimensions

%Check Arguments
if nargin<4
aspect = [0.3756 0.3756 1] ;
end
if nargin<3
error('Not enough input arguments')
end

aspectRatio = aspect(1)/aspect(3);
sz = size(dapiImg);
% Load Points
%
[ignore1,pX,pY,pZ,ignore2] = textread(pts_fname,'%d%d%d%d%d');
clear pts_fname ignore1 ignore2
pZ = pZ - 1; % Poinpicker take first slice as 1 instead of 0
mask1 = newim(sz);
for ii=1:length(pX)
mask1(pX(ii),pY(ii),pZ(ii)) = 1;
end
mask1 = mask1 == 1;
nucleiCoords = dip_image(mask1,'uint16');
clear mask1;
nucleiCoords = label(nucleiCoords);
msrA = measure(nucleiCoords,[],'center');
no_nuc = size(msrA,1);
no_nuc
id = msrA.id;
maxid = max(id);
lab = zeros(1,maxid);
lab(id) = id;
lab = uint16([0,lab]);
nucleiCoords = lut(nucleiCoords,lab);
clear lab id maxid msrA

%Model Nuclei
size_of_parabolic_kernel = 30;
tmp = ~dapiSignal;
dapiImg_model = dapiImg;
%clear dapiImg
dapiImg_model(tmp) = 4095;
offset = erosion(dapiImg_model,size_of_parabolic_kernel*[1,1,aspectRatio],'parabolic');
offset = offset(dapiSignal);
dapiImg_model(tmp) = 0;
clear tmp
range = dilation(dapiImg_model,size_of_parabolic_kernel*[1,1,aspectRatio],'parabolic');
range = max(range(dapiSignal)-offset,1);
tmp = (dapiImg_model(dapiSignal)-offset)/range;
clear offset range

```

```

tmp = 1-clip(tmp,1,0);
dapiImg_model(dapiSignal) = exp(3*tmp);
clear tmp size_of_parabolic_kernel

% Find the full nuclei
modelledNuclei = nucleiCoords;
modelledNuclei = dip_growregionsweighted(modelledNuclei,dapiImg_model,dapiSignal,aspect,5,[]);
clear ogm aspect dapiImg_model

```

pdu_find_model_Centers.m

```
function [modelledNuclei,no_nuc] = pdu_find_model_Centers(pts,dapiImg,dapiSignal,aspect)
```

```

%Check Arguments
if nargin<4
    aspect = [0.3756 0.3756 1] ;
end
if nargin<3
    error('Not enough input arguments')
end

aspectRatio = aspect(1)/aspect(3);
sz = size(dapiImg);

pX = pts(1,:);
pY = pts(2,:);
pZ = pts(3,:);

mask1 = newim(sz);
for ii=1:length(pX)
    mask1(pX(ii),pY(ii),pZ(ii)) = 1;
end
mask1 = mask1 == 1;
nucleiCoords = dip_image(mask1,'uint16');
clear mask1;
nucleiCoords = label(nucleiCoords);
msrA = measure(nucleiCoords,[],'center');
no_nuc = size(msrA,1);
no_nuc
id = msrA.id;
maxid = max(id);
lab = zeros(1,maxid);
lab(id) = id;
lab = uint16([0,lab]);
nucleiCoords = lut(nucleiCoords,lab);
clear lab id maxid msrA

% Model Nuclei
size_of_parabolic_kernel = 30;
tmp = ~dapiSignal;
dapiImg_model = dapiImg;
%clear dapiImg
dapiImg_model(tmp) = 4095;
offset = erosion(dapiImg_model,size_of_parabolic_kernel*[1,1,aspectRatio],'parabolic');
offset = offset(dapiSignal);
dapiImg_model(tmp) = 0;
clear tmp
range = dilation(dapiImg_model,size_of_parabolic_kernel*[1,1,aspectRatio],'parabolic');
range = max(range(dapiSignal)-offset,1);
tmp = (dapiImg_model(dapiSignal)-offset)/range;
clear offset range
tmp = 1-clip(tmp,1,0);
dapiImg_model(dapiSignal) = exp(3*tmp);
clear tmp size_of_parabolic_kernel

% Find the full nucleiCoords
modelledNuclei = nucleiCoords;

```

```
modelledNuclei = dip_growregionsweighted(modelledNuclei,dapiImg_model,dapiSignal,aspect,5,[]);
clear ogm aspect dapiImg_model
```

write_centerCord_in_pointpicker.m

```
function [no_nuc] = write_centerCord_in_pointpicker(centerCords,cfname)

cX = centerCords(1,:);
cY = centerCords(2,:);
cZ = centerCords(3,:);

[cZ_sort,I] = sort(cZ);
clear cZ_sort
cZ = cZ(I);
cY = cY(I);
cX = cX(I);
cZ = cZ + 1;
cZ = round(cZ);
cY = round(cY);
cX = round(cX);

no_nuc = length(cZ);

fid = fopen(cfname,'w');
fprintf(fid,'%s %s %s %s %s\n','point','x','y','slice','color');
prev_id = cZ(1);
count_id = -1 ;

for ii = 1:length(cZ)
    if cZ(ii) == prev_id
        count_id = count_id + 1;
    else
        count_id = 0;
    end
    prev_id = cZ(ii);
    fprintf(fid,'%d %d %d %d %d\n',[count_id cX(ii) cY(ii) cZ(ii) count_id]);
end

fclose(fid);
no_nuc
```

pdu_find_model_rm_small_nuc.m

```
function [fnuclei,no_nuc] = pdu_find_model_rm_small_nuc(pts_fname,dapiImg,dapiSignal,aspect,lower_sz_thresh)

%Check Arguments
if nargin<4
    aspect = [0.3756 0.3756 1] ;
end
if nargin<3
    error('Not enough input arguments')
end

aspectRatio = aspect(1)/aspect(3);
sz = size(dapiImg);

[ignor1,pX,pY,pZ,ignore2] = textread(pts_fname,'%d%d%d%d%d');
clear pts_fname ignor1 ignore2
pZ = pZ - 1; % Poinpicker take first slice as 1 instead of 0
mask1 = newim(sz);
for ii=1:length(pX)
    mask1(pX(ii),pY(ii),pZ(ii)) = 1;
end
mask1 = mask1 == 1;
nucleiCoords = dip_image(mask1,'uint16');
clear mask1;
nucleiCoords = label(nucleiCoords);
```

```

msrA = measure(nucleiCoords,[],'center');
no_nuc = size(msrA,1);
no_nuc
id = msrA.id;
maxid = max(id);
lab = zeros(1,maxid);
lab(id) = id;
lab = uint16([0,lab]);
nucleiCoords = lut(nucleiCoords,lab);
clear lab id maxid msrA

%Model Nuclei
size_of_parabolic_kernel = 30;
tmp = ~dapiSignal;
dapiImg_model = dapiImg;
%clear dapiImg
dapiImg_model(tmp) = 4095;
offset = erosion(dapiImg_model,size_of_parabolic_kernel*[1,1,aspectRatio],'parabolic');
offset = offset(dapiSignal);
dapiImg_model(tmp) = 0;
clear tmp
range = dilation(dapiImg_model,size_of_parabolic_kernel*[1,1,aspectRatio],'parabolic');
range = max(range(dapiSignal)-offset,1); % Avoid division by zero.
tmp = (dapiImg_model(dapiSignal)-offset)/range;
clear offset range
tmp = 1-clip(tmp,1,0);
dapiImg_model(dapiSignal) = exp(3*tmp);
clear tmp size_of_parabolic_kernel

% Find the full nucleiCoords
modelledNuclei = nucleiCoords;
modelledNuclei = dip_growregionsweighted(modelledNuclei,dapiImg_model,dapiSignal,aspect,5,[]);

msr = measure(modelledNuclei,[],('size','center'));
to_rm = msr.size < lower_sz_thresh;
id = msr.id ;
for ii = find(to_rm)
    nucleiCoords(nucleiCoords == id(ii)) = 0;
end

mask1 = nucleiCoords > 0;
nucleiCoords = dip_image(mask1,'uint16');
clear mask1;
nucleiCoords = label(nucleiCoords);
msrA = measure(nucleiCoords,[],'center');
no_nuc = size(msrA,1);
no_nuc
id = msrA.id;
maxid = max(id);
lab = zeros(1,maxid);
lab(id) = id;
lab = uint16([0,lab]);
nucleiCoords = lut(nucleiCoords,lab);
clear lab id maxid msrA
modelledNuclei = nucleiCoords;
modelledNuclei = dip_growregionsweighted(modelledNuclei,dapiImg_model,dapiSignal,aspect,5,[]);
clear ogm aspect dapiImg_model

```

6.1.4 Expression to model assignment program

The code is written in MATLAB and requires DIPlib (www.diplib.org) for functionality. This program works as follows: reads spatial average expression patterns, superimpose them onto the cellular model, calculate the mean intensity in each of the cell, normalize

these mean values to generate Z-scores and put a threshold on the z-score to generate 0 or 1 for no expression or expression.

expression_assignment_Zscore_based_highThrough_varZThresh.m

```
function expression_assignment_Zscore_based_highThrough_varZThresh(data_dir,threshFileName)

mat_out_fileName = [data_dir 'Expression-mat_' date '.txt'];
zscore_mat_out_fileName = [data_dir 'Expression-zscore_' date '.txt'];

tic

% Write the parameters used
fid3 = fopen([zscore_mat_out_fileName '_' 'log.txt'],'w');

%Input parameters
fprintf(fid3,'PWD=%s\n',pwd);
fprintf(fid3,'%s\n',expression_assignment_Zscore_based_highThrough(data_dir,mat_out_fileName,zscore_mat_out_fileName,Zscore
_thresh));
fprintf(fid3,'data_dir=%s\n',data_dir);
fprintf(fid3,'mat_out_fileName=%s\n',mat_out_fileName);
fprintf(fid3,'zscore_mat_out_fileName=%s\n',zscore_mat_out_fileName);

%Aspect
aspect = [0.3756 0.3756 1];
ar = aspect(1)/aspect(3);

% New Average Model
fnuclei_fileName = 'C:\Documents and Settings\tomer\My
Documents\MATLAB\NuclearModelling48hpf\Final_Individual_Model_To_Use\Model_100608_2_fnuclei1_200thresh.ics';
fnuclei = readim(fnuclei_fileName,'ICS');

fprintf(fid3,'fnuclei_fileName: %s\n',fnuclei_fileName);

% to find no. of modelled nuclei
msr = measure(fnuclei,[],{'size'});
no_cells = sum(msr.size > 0);
clear msr

fprintf(fid3,'no_cells: %d\n',no_cells);

%Import threshold data
thresh_data = importdata(threshFileName);
thresh_data_geneIDs = thresh_data.rowheaders;
thresh_data = thresh_data.data;

% Data directory parsing
im_data_file = dir([data_dir '*.PIC']);
sz = size(im_data_file);
no_ims = sz(1); % no of images in the data dir
disp(sprintf('\n\nData Dir=%s',data_dir));
disp(sprintf('\n\nNo. of Images being used:%d',no_ims));
fprintf(fid3,'Input Data Dir used=%s\n',data_dir);
fprintf(fid3,'No. of Images being used:%d\n',no_ims);

% Data Matrix declaration
exp_data_mat = zeros(no_cells,no_ims,'int8');
zscore_mat = zeros(no_cells,no_ims,'double');
gene_info = "";

% Z-score threshold is defined here

factor_for_mean_of_geneExp = 8; % Magic no.
fprintf(fid3,'factor_for_mean_of_geneExp:%d\n',factor_for_mean_of_geneExp);

for ii = 1:no_ims % for all the images
    tic
        zscore_thresh_index = -1;
```

```

for zz = 1:length(thresh_data)
    tryMatch = regexp(im_data_file(ii).name,thresh_data_geneIDs{zz});
    if (tryMatch == 1)
        zscore_thresh_index = zz;
        break
    end
end

disp(sprintf('Matched File Name %s and Gene ID %s',im_data_file(ii).name,thresh_data_geneIDs{zscore_thresh_index}));
fprintf(fid3,'\nMatched File Name %s and Gene ID %s',im_data_file(ii).name,thresh_data_geneIDs{zscore_thresh_index});

Zscore_thresh = thresh_data(zscore_thresh_index);
disp(sprintf('\nZscore Threshold to use:%.2g',Zscore_thresh));
fprintf(fid3,'\nZscore Threshold to use:%.2g',Zscore_thresh);
fprintf(fid3,'\nParameters for:%s, Image No. %d out of total %d\n',im_data_file(ii).name,ii,no_ims);
fprintf(fid3,'%s\n','-----');
gene_no = ii;
gene_info = strvcat(gene_info,im_data_file(ii).name); % saving gene IDs
fname = [data_dir im_data_file(ii).name];
disp(sprintf('\nAnalyzing %s',im_data_file(ii).name));
disp(sprintf('\nImage no:%d out of total:%d',ii,no_ims));
geneExp = readim(fname,'PIC');
[thresh_mask,thres] = threshold(geneExp,'isodata');
thres
fprintf(fid3,'Gaussian smothering used with the function: %s with ar=%2.5g\n','gaussf_iir(geneExp,2*[1 1 ar]),ar);
msr = measure(fnuclei.geneExp,({'size','center','mean','sum','stddev'}));
posIds = msr.mean > thres/100; % IDs with more than certain threshold
no_of_nuclei_not_considered = no_cells - sum(posIds);
disp(sprintf('No. of Nuclei not included:%d out of Total:%d,no_of_nuclei_not_considered,no_cells));
fprintf(fid3,'No. of Nuclei not included:%d out of Total:%d\n',no_of_nuclei_not_considered,no_cells);
mean1 = mean(msr.mean(posIds));
std1 = std(msr.mean(posIds));
zsc1 = (msr.mean - mean1)/std1; % Zscore is calculated here
[tmp,thres_zscore] = threshold(zsc1,'isodata');thres_zscore
clear tmp
zsc1 = zsc1 + 1 - thres_zscore; %Normalize z-score with the threshold so that the threshold to use is 1
zscore_mat(:,ii) = zsc1;
ids = msr.id;
positive_cells = zsc1 > Zscore_thresh; % cells which have signal mean more than a zscore cutoff
exp_data_mat(:,ii) = positive_cells;
fprintf(fid3,'Zscore_thresh:%2.6g\n',Zscore_thresh);
disp(sprintf('No. of positive Cells %d',sum(positive_cells)));
fprintf(fid3,'No. of positive Cells %d\n',sum(positive_cells));
nucs_retained = fnuclei; % Empty images are defined for nucs

maxid = max(msr.id);
lab = zeros(1,maxid);
lab(msr.id) = 0;
lab1 = lab;
lab2 = lab;
clear lab
lab1(msr.id(positive_cells)) = 255;
lab1 = uint16([0,lab1]);
lab2(msr.id(positive_cells)) = msr.id(positive_cells);
lab2 = uint16([0,lab2]);
nucs_retained = lut(fnuclei,lab1);
nucs_retained_lab = fnuclei;
nucs_retained_lab = lut(fnuclei,lab2);

nucs_retained = dip_image(nucs_retained,'uint8');
nucs_retained_fileName = [fname '_nucs_retained_ZscoreCut' num2str(Zscore_thresh)];
nucs_retained_lab_fileName = [fname '_nucs_retained_lab_ZscoreCut' num2str(Zscore_thresh)];
writeim(nucs_retained,nucs_retained_fileName,'ICSv2',0,1);
writeim(nucs_retained_lab,nucs_retained_lab_fileName,'ICSv2',0,1);
fprintf(fid3,'File Written %s\n',[nucs_retained_fileName '.ics']);
fprintf(fid3,'File Written %s\n',[nucs_retained_lab_fileName '.ics']);
toc
end

```

```

% Write the Matrix and Zscore out
fid = fopen(mat_out_fileName,'w');
fid2 = fopen(zscore_mat_out_fileName,'w');
fprintf(fid,'%s','UNIQID');
fprintf(fid2,'%s','UNIQID');

for n = 1:no_ims
    fprintf(fid,'\t%s',gene_info(n,:));
    fprintf(fid2,'\t%s',gene_info(n,:));
end
fprintf(fid,'\n');
fprintf(fid2,'\n');
for m = 1:no_cells
    fprintf(fid,'%d',ids(m));
    fprintf(fid2,'%d',ids(m));
    for n = 1:no_ims
        fprintf(fid,'\t%d',exp_data_mat(m,n));
        fprintf(fid2,'\t%.8g',zscore_mat(m,n));
    end
    fprintf(fid,'\n');
    fprintf(fid2,'\n');
end
fclose(fid);
fclose(fid2);

total_time_taken = toc;
disp(sprintf('Total time taken %9.9g Seconds!',total_time_taken));
fprintf(fid3,'Total time taken %9.9g Seconds!\n',total_time_taken);
fclose(fid3); % Parameters file close

```

6.1.5 *mfpBLAST*

The code is written in MATLAB and requires DIPlib (www.diplib.org) for functionality.

mfpBLAST.m

```

function varargout = mfpBLAST(varargin)

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @mfpBLAST_OpeningFcn, ...
    'gui_OutputFcn', @mfpBLAST_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before mfpBLAST is made visible.
function mfpBLAST_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to mfpBLAST (see VARARGIN)

```

```

% Read the matrix data here
% Populate the List of genes

[list_of_genes] = readMatFile();

handles.list_of_genes = list_of_genes;
handles.current_gene = handles.list_of_genes{2};
handles.present_genes = {};
handles.not_present_genes = {};
set(handles.availableGenes,'String',handles.list_of_genes);

% Choose default command line output for mfpBLAST
handles.output = hObject;

% Update handles structure
guidata(handles.availableGenes,handles);
guidata(hObject, handles);

% UIWAIT makes mfpBLAST wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = mfpBLAST_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in availableGenes.
function availableGenes_Callback(hObject, eventdata, handles)
% hObject handle to availableGenes (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
val = get(hObject,'Value');
str = get(hObject,'String');
handles.current_gene = str{val};
guidata(hObject,handles);

% Hints: contents = get(hObject,'String') returns availableGenes contents as cell array
% contents{get(hObject,'Value')} returns selected item from availableGenes

% --- Executes during object creation, after setting all properties.
function availableGenes_CreateFcn(hObject, eventdata, handles)
% hObject handle to availableGenes (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
% See ISPC and COMPUTER.
%set(hObject,'String',handles.list_of_genes);

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%set(hObject,'String',get(handles.mfpBLAST,list_of_genes));
guidata(hObject,handles);

% --- Executes on button press in add_to_presentList.
function add_to_presentList_Callback(hObject, eventdata, handles)
% hObject handle to add_to_presentList (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

str = get(handles.availableGenes,'String');
val = get(handles.availableGenes,'Value');
sz = length(handles.present_genes);
handles.present_genes{sz+1} = str{val};
set(handles.genesPresentListBox,'String',handles.present_genes)
guidata(handles.genesPresentListBox,handles)
guidata(hObject,handles);

% --- Executes on button press in add_to_notPresentList.
function add_to_notPresentList_Callback(hObject, eventdata, handles)
% hObject handle to add_to_notPresentList (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
str = get(handles.availableGenes,'String');
val = get(handles.availableGenes,'Value');
sz = length(handles.not_present_genes);
handles.not_present_genes{sz+1} = str{val};
set(handles.genesNotPresentListBox,'String',handles.not_present_genes)
guidata(handles.genesNotPresentListBox,handles)
guidata(hObject,handles);

% --- Executes on selection change in genesPresentListBox.
function genesPresentListBox_Callback(hObject, eventdata, handles)
% hObject handle to genesPresentListBox (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%set(hObject,'String',handles.present_genes);

set(hObject,'String',handles.present_genes);

guidata(hObject,handles);
% Hints: contents = get(hObject,'String') returns genesPresentListBox contents as cell array
% contents {get(hObject,'Value')} returns selected item from genesPresentListBox

% --- Executes during object creation, after setting all properties.
function genesPresentListBox_CreateFcn(hObject, eventdata, handles)
% hObject handle to genesPresentListBox (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in genesNotPresentListBox.
function genesNotPresentListBox_Callback(hObject, eventdata, handles)
% hObject handle to genesNotPresentListBox (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
guidata(hObject,handles);
% Hints: contents = get(hObject,'String') returns genesNotPresentListBox contents as cell array
% contents {get(hObject,'Value')} returns selected item from genesNotPresentListBox

% --- Executes during object creation, after setting all properties.
function genesNotPresentListBox_CreateFcn(hObject, eventdata, handles)
% hObject handle to genesNotPresentListBox (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.

```

```

% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in mfpBLAST.
function mfpBLAST_Callback(hObject, eventdata, handles)
% hObject handle to mfpBLAST (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
present = handles.present_genes;
not_present = handles.not_present_genes;
mpfBlastImpl(present,not_present);

% --- Executes on button press in remove_from_presentList.
function remove_from_presentList_Callback(hObject, eventdata, handles)
% hObject handle to remove_from_presentList (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%str = get(handles.genesPresentListBox,'String');
val = get(handles.genesPresentListBox,'Value');
sz = length(handles.present_genes);
handles.present_genes{val} = [];

if (val ~= sz)
temp_present_genes = {};
jj = 0;
for ii = 1:sz
    if (ii ~= val)
        jj = jj + 1;
        temp_present_genes{jj} = handles.present_genes{ii};
    end
end

handles.present_genes = temp_present_genes;
end

%handles.present_genes = unique(handles.present_genes);
set(handles.genesPresentListBox,'String',handles.present_genes)
guidata(handles.genesPresentListBox,handles)
guidata(hObject,handles);

% --- Executes on button press in remove_From_notPresentList.
function remove_From_notPresentList_Callback(hObject, eventdata, handles)
% hObject handle to remove_From_notPresentList (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
val = get(handles.genesNotPresentListBox,'Value');
sz = length(handles.not_present_genes);
handles.not_present_genes{val} = [];

if (val ~= sz)
temp_present_genes = {};
jj = 0;
for ii = 1:sz
    if (ii ~= val)
        jj = jj + 1;
        temp_present_genes{jj} = handles.not_present_genes{ii};
    end
end

handles.not_present_genes = temp_present_genes;
end

%handles.present_genes = unique(handles.present_genes);
set(handles.genesNotPresentListBox,'String',handles.not_present_genes)
guidata(handles.genesNotPresentListBox,handles)
guidata(hObject,handles);

```

```

%-----Non-GUI Functions-----
%-----

function [list_of_genes ] = readMatFile()
exp_mat_fileName = 'C:\tomer\softwares\MATLAB7\work\Data\Expression-mat_18-Jun-2008_ver62_IDS_corrected-
ver8_PhDAnalysis.txt' ;
expression_data_read = importdata(exp_mat_fileName);
%expression_mat_data = expression_data_read.data;
list_of_genes = expression_data_read.colheaders;

clear expression_data_read exp_mat_fileName

function [nucs_labelled] = mpfBlastImpl(present,not_present)

if (length(present) == 0 & length(not_present) == 0)
    error {'Atleast one of Present or Not Present Cell array should be assigned'}
end

% Matrix Data
mat_file = 'C:\tomer\softwares\MATLAB7\work\Data\Expression-mat_18-Jun-2008_ver62_IDS_corrected-ver8_PhDAnalysis.txt' ;
data_import = importdata(mat_file);
data_mat = data_import.data ;
sz = size(data_mat);

%Read Nuc Model
fnuclei_fileName = 'C:\tomer\softwares\MATLAB7\work\Data\Model_100608_2_fnuclei1_200thresh.ics';
disp(sprintf('Reading Model Data from %s',fnuclei_fileName));
fnuclei = readim(fnuclei_fileName,'ICS');
clear fnuclei_fileName

gene_IDS = data_import.colheaders;
clear data_import
%remove null entries from present and not_present

if length(present) > 0
    temp = {};
    jj = 0;
    for ii = 1:length(present)
        if (length(present{ii}) ~= 0)
            jj = jj + 1;
            temp{jj} = present{ii};
        end
    end
    present = temp;
end

if length(not_present) > 0
    temp = {};
    jj = 0;
    for ii = 1:length(not_present)
        if (length(not_present{ii}) ~= 0)
            jj = jj + 1;
            temp{jj} = not_present{ii};
        end
    end
    not_present = temp;
end

%indices for present analysis
present_ind = zeros(1,length(present));

isfound = 0;

if (length(present) > 0)

for ii = 1:length(present)

```

```

for jj = 1:length(gene_IDs)
    match = regexp(gene_IDs(jj),[present{ii}]);
    if sum(match{1}) > 0
        present_ind(ii) = jj;
        isfound = isfound + 1;
        break
    end
end
if (isfound == 0)
    present(ii)
    error {'Gene Not Found'}
end
if (isfound > 1)
    present(ii)
    error {'Multiple Genes Found'}
end
isfound = 0;
end
present_mask = sum(data_mat(:,present_ind))' > length(present_ind) - 1;
if (length(present) == 1)
    present_mask = data_mat(:,present_ind);
end

end

%indices for not present analysis
if (length(not_present) > 0)
    not_present_ind = zeros(1,length(not_present));
    isfound = 0;
    for ii = 1:length(not_present)
        for jj = 1:length(gene_IDs)
            match = regexp(gene_IDs(jj),[not_present{ii}]);
            if sum(match{1}) > 0
                not_present_ind(ii) = jj;
                isfound = isfound + 1;
                break
            end
        end
    end
    if (isfound == 0)
        not_present(ii)
        error {'Gene Not Found'}
    end
    if (isfound > 1)
        not_present(ii)
        error {'Multiple Genes Found'}
    end
    isfound = 0;
end

not_present_mask = (sum(data_mat(:,not_present_ind))' > 0;

if (length(not_present) == 1)
    not_present_mask = data_mat(:,not_present_ind);
end

end

if (length(present) > 0 & length(not_present) > 0)
    final_mask = present_mask > not_present_mask;
end

if (length(present) > 0 & length(not_present) == 0)
    final_mask = present_mask > 0;
end

if (length(present) == 0 & length(not_present) > 0)
    final_mask = not_present_mask < 1;
end

```

```

msr = measure(fnuclei,[],{'size'});
maxid = max(msr.id);
lab = zeros(1,maxid);
lab(msr.id) = 0;
lab_ids = data_mat(final_mask,1);
lab(lab_ids) = lab_ids;
clear lab_ids
lab = uint16([0,lab]);

disp(sprintf('Assigning Data to variable name:%s','nucs_labelled'));
nucs_labelled = lut(fnuclei,lab);

%read tubulin

avgTubFileName = 'C:\tomer\softwares\MATLAB7\work\Data\ver5Avg.PIC';
avgTub = readim(avgTubFileName,'PIC');
%nucs_labelled
nucs_labelled_overlayed = overlay(avgTub,nucs_labelled)
nucs_labelled_overlayed
dipanimate(gcf,'loop')
%dipshow(nucs_labelled,'labels')
%overlay(avgTub,nucs_labelled)
%close()
%nucs_labelled_overlayed

```

6.1.6 PduBrainExplorer

The code is written in Java as a plugin in ImageJ (<http://rsbweb.nih.gov/ij/>)

```

import ij.IJ;
import ij.ImagePlus;
import ij.ImageStack;
import ij.gui.ImageCanvas;
import ij.gui.ImageWindow;
import ij.plugin.filter.PlugInFilter;
import ij.process.ImageProcessor;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.*;
import java.util.HashMap;
import java.util.Map;
import javax.swing.text.AbstractDocument.BranchElement;
import com.sun.org.apache.xpath.internal.operations.Bool;

public class cellTypes_Explorer implements MouseListener, PlugInFilter {

    private ImagePlus img;
    private ImageCanvas canvas;
    private ImageStack imgStack;
    private Map<Integer,Boolean[]> cTypesHashMap = new HashMap<Integer,Boolean[]>();
    public String[] geneIDs;

    @Override
    public void mouseClicked(MouseEvent e) {
        int x = e.getX();
        int y = e.getY();
        int offscreenX = canvas.offScreenX(x);
        int offscreenY = canvas.offScreenY(y);
        int[] cTypesId = img.getPixel(offscreenX, offscreenY);
        this.printExpressedGenesList(cTypesId[0]);
        //IJ.showMessage("Mousepressed: "+offscreenX+","+offscreenY+" GreyValue ="
+cTypesId[0]);

```

```

}

public void loadCTTypesDatabase(String matFileName) {

    BufferedReader br = null;
    try{
        br = new BufferedReader(new FileReader(matFileName));
        String line = null;
        line = br.readLine();
        geneIDs = line.split("\t");
        System.out.println(geneIDs[0]);
        while ((line = br.readLine()) != null){
            String [] temp_string = line.split("\t");
            Boolean [] temp_bool = new Boolean[temp_string.length-1];
            for (int i = 1; i < temp_string.length; i++){
                if (temp_string[i].equals("0")) temp_bool[i-1] = false;
                if (temp_string[i].equals("1")) temp_bool[i-1] = true;
                //System.out.println(temp_bool[i-1]);
            }
            cTypesHashMap.put(new
Integer(Integer.parseInt(temp_string[0])),temp_bool);
            //System.out.println((cTypesHashMap.get(new Integer(1))[1]);
        }
    }
    catch (IOException ex)
    {
        System.out.println("File Not Found");
    }
}

public void printExpressedGenesList(int cTId){
    Boolean[] temp_bool = (Boolean[])cTypesHashMap.get(new Integer(cTId));
    //System.out.println(geneIDs[1]);
    //IJ.log(""+geneIDs.toString());
    //IJ.log(geneIDs[1]+"in");
    IJ.setColumnHeadings("");
    IJ.write("Genes Present in Cell ID: CT" + cTId);
    IJ.write("-----");
    for (int i = 0; i < temp_bool.length; i++){
        //IJ.showMessage(geneIDs[i+1]+"in");

        if (temp_bool[i]){
            IJ.write(geneIDs[i+1]);
            //System.out.println(geneIDs[i+1]);
        }
    }
}

@Override
public void mouseEntered(MouseEvent e) {
}

@Override
public void mouseExited(MouseEvent arg0) {
}

@Override
public void mousePressed(MouseEvent e) {
    mouseClicked(e);
}

@Override
public void mouseReleased(MouseEvent arg0) {
}

@Override
public void run(ImageProcessor ip) {
    String cTypesMatFileName = "current-mat.txt";
    this.loadCTTypesDatabase(cTypesMatFileName);
    ImageWindow win = img.getWindow();
    canvas = win.getCanvas();
}

```

```

        canvas.addMouseListener(this);
    }

    @Override
    public int setup(String arg, ImagePlus imp) {
        this.img = imp;
        return NO_CHANGES + DOES_ALL;
    }

    /**
     * @param args
     */
    public static void main(String[] args) {
        String cTypesMatFileName = "current-mat.txt";
        System.out.println(cTypesMatFileName);
        cellTypes_Explorer ct = new cellTypes_Explorer();
        ct.loadCTypesDatabase(cTypesMatFileName);
        ct.printExpressedGenesList(1100);
    }

```

6.2 Sequences and trees

6.2.1 *Dach* Sequence and tree

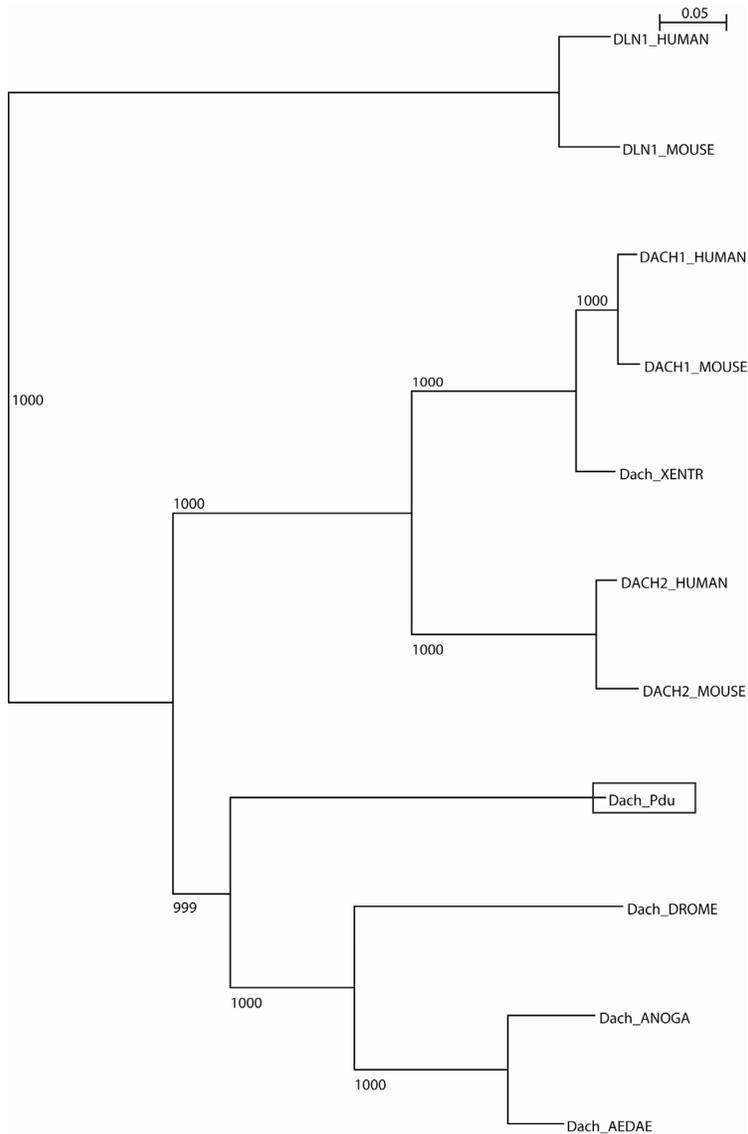
>PduDach

```

GCTTGGTGGAGCCCCAGTCGCATTCATCCCATTTCAGAGAAGAAGGAGAAGCACCTGGCAC
TCACAACCTGGACCAGTCGCTCTCTGGTCTTCATCTCTCTCTGTTGTGTTACTGATTGC
AAAAGCGTCACATTGTTGTCTATTGACTTTTTTACAAGACGTTATTACAACCTGGATAGCTC
ATAACTCGGACTGTCAGCACTTGCTGAGGAGTATACACTTGCATTTGAGACTTCTTCACA
AACTTTGGACTGACTTTACAGCGAGTTGACACACAAGTTGGGCTGGCTTTTTATTCACTT
TTATCTGCCAAGATGATGATGGAAGCAGTTCCCCCGGGCCACCTCCTTACCATCGCCC
AGCCGGACTCCGGTGTGAGTCCCTCAGCCGCCCCCGGCCCTATGCACCAAGGGCCCCC
TTGCCCTCCCCCTCCCTCGCACTCCATTGTGTCTATGTCCTCAATCTCAACCACGACCTCC
AGCATCCTCCCCCTAAAGATGGAGAAGCCCATCTACTCATCCCCCTCCCCCGCAGCCTCG
AACCCCGAGAACAACACTTGCAAAATGATCGACTACCGAGGGGCCAAGGTGGCCGCCTTC
AAGGTGGATGGCCGCGAGCTCATCTGCCTGCCCCAGGCCTTCGAGCTCTTCCTCAAGCAC
CTGGTTCGGAGGGCTGCACACCGTCTACACCAAACCAAGAGACTGGACATTACTCCCATC
GTGTGCAATGTGGAGCAGGTCCGGATCCTCCGGGGGCTCGGGGCCATCCAGCCGGGAGTG
AACCGTGCAAACTCATCTCCCCCGGGGAGTTCGACGTGCTCTATGACGACTGCACCAAC
TCAAGTGTCTCGGCCCGCAGACCCCCAAGAGAAGCCCCATTGCAGCACACCCGGAGACA
ATCGAGAAGCTGAAGAAACAGAGGATGGAGGGGGAGTACCCTTATTACCCAACAGATTA
TTAGGTCAAGTTCATGTTCTCAGACCCCAAGAAGTCTCCCCTGTTTCGGCAATGGCTACCAC
TACCCCCCTCACCTGGCCTCGATGAATGTGCCCTTCATGCCCTTAGGACATCCCATGATG
ACGATGGCCATGGCTAACACATGGGAATGAGGCCTGACGGATCCATCATCAGGGAAAGA
CAGGCCCCCCGAGGACCTTGCAATCCCCCAGACCCAAAGATGATGGCAGATATGACGAGAAC
AACATGGAACACAACAACAACAACCCCTGGACAAACCCCTCAATCTGCAGATGGACAGG
CCGAGGGAGAGAAGACCCTCACTGACCGACAAAGATTTCAGGAATGAGCACCCCCAGTGAC
GTCATGACCAACGGCCAATTAGATTTGTCCATGAAGCACTCAGCGCCATCTATCAACGAG
GATCTGGATGACGACTCAGAAGATGACAAGGAGGACGACGATGATATGGACGACTCAGAT
GCCCCCTCAAGCATGGCCAACAGTGATGCCGCTGACAAGATGGCCGCCTCCTCCCTGCC
TACCAAGCTTCGCAGATGCTCAACACTGAGACCACTGGCATCTCGTCGGTGGAGACATTG
TTGATGAACATCCAAGGGCTCCTCAAGGTTCGCTCAGAAAACGCCAGGCACAGGGAACGC
CAGATGAACTACGAGAAAGCTGAACTCAAGATGGAAGTATGAGAGAAAGAGAGTTGAGA
GAAAGTCTGGAGAAGCAATGGGAGATGAGCAGAGAACAAGGTCAATCTGCAAAGGAGG
TAAAAAAAAAAAAA

```

Neighbour joining tree of *Dach* with 1000 bootstrap



6.2.2 *Svp/COUPTF1* Sequence and tree

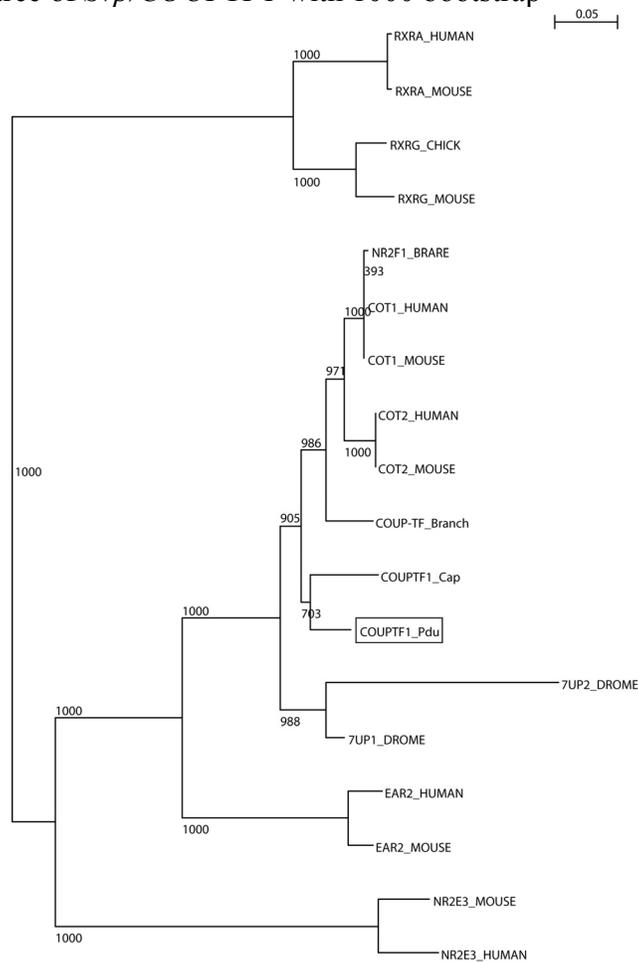
>Pdu_COUPTF1

```

CAAGAGCTCGGGCAAGCATTATGGCCAGTTTACCTGCGAGGGGCTGCAAGAGTTTCTTCAA
GCGATCGGTGAGGAGGAACCTGACCTACACATGCAGGGGCAACAGGACGTGTCCCATCGA
CCAGCACCACAGGAACCAGTGTCAATACTGCAGGCTCAAGAAATGCCTCAAGATGGGCAT
GAGGAGAGAAGCCGTGCAGAGGGGGAGAGTGCCTCCAACCCAGCACCCAGGCTTCCCAGG
ACAGATGCTTGCCAATGGAGACCCCCCTGAATGGACACACATATCTATCCAGCTTCATCTC
CATGCTGCTGAGAGCCGAGCCCTACCCACGTCAAGATATGGACAGTGCATGCAACCCAA
CAACATCATGGGCATCGAGAATATCTGTGAACTGGCAGCCAGGCTGCTCTTCAGTGCTGT
CGAGTGGTCAAGGAACATCCCCTTCTTCCCTGACCTCCAAGTAACTGACCAAGTTGCCCT
CCTCAGACTCAGTTGGAGTGAAGTGTGGTCTCAACGCTGCCAGTGCCTCTATGCCCTT
ACATGTGGCACCCTTACTGGCTGCTGCAGGGCTCCATGCCAGTCCCATGGCTGCGGACAG
AGTAGTGGCTTTCATGGATCATATACGTATTTTCCAAGAACAAGTAGAAAAACTGAAAGC
CTTGCACGTCGACTCTGCGGAATATAGTTGTCTAAAGGCTGTAGTATTATTCTCATCAGA
TGCCTGTGGATTGTGCGACACCGCTCACATCGAGAGTCTACAGGAGAAGAGCCAGTGC GC
CCTGGAGGAGTACGTTTCGGAGCCAGTACCCCAACCAGCCGACCAGGTTTCGGAAAAC TGCT
GCTCCGGTTGCCCTCGCTGAGGTCCGTGTCTGCCAGGTCATCGAACAACTCTTCTTCGT
ACGACTC

```

Neighbour joining tree of *Svp/COUPTF1* with 1000 bootstrap



6.2.3 Table of *Platynereis* Transcription Factors candidates identified in the screen

IDs	IBO_Ids	Plate_No	Coords	BLAST First Best Hit, Short description	BLAST First Best Hit
H1	IB0AAA15BC06EM1	15	E12	Cell division cycle 5-like protein - Homo sap...	Q99459 CDC5L_HUMAN&392&e-108
H2	IB0AAA15CC12EM1	15	F23	Protein DEK - Homo sapiens (Human)	P35659 DEK_HUMAN&90&1e-17
H3	IB0AAA15CD03EM1	15	H5	Thyroid transcription factor 1 - Homo sapiens...	P43699 TTF1_HUMAN&177&8e-44
H4	IB0AAA16AE02EM1	16	I3	cAMP-responsive element modulator - Canis fami...	P79145 CREM_CANFA&119&3e-26
H5	IB0AAA16DB01EM1	16	D21	Transcription factor Sp8 - Mus musculus (Mouse)	Q8BMJ8 SP8_MOUSE&181&7e-58
H6	IB0AAA17CD06EM1	17	H11	Four and a half LIM domains protein 2 - Homo s...	Q14192 FHL2_HUMAN&357&5e-98
H7	IB0AAA17CG03EM1	17	N5	Class B basic helix-loop-helix protein 5 - Me...	O09029 BHLH5_MESAU&132&2e-30
H8	IB0AAA18DA04EM1	18	B8	Oligodendrocyte transcription factor 3 - Mus ...	Q6PFG8 OLIG3_MOUSE&121&6e-27
H9	IB0AAA19AG03FM1	19	M5	Transcriptional repressor p66 alpha - Homo sap...	Q86YP4 P66A_HUMAN&117&7e-26
H10	IB0AAA19BF01EM1	19	K2	Protein max - Rattus norvegicus (Rat)	P52164 MAX_RAT&135&2e-31
H11	IB0AAA19CF09EM1	19	L17	Zinc finger protein ubi-d4 - Gallus gallus (Ch...	P58268 REQU_CHICK&97&2e-19
H12	IB0AAA19DF09EM1	19	L18	Aryl hydrocarbon receptor nuclear translocator...	O15945 ARNT_DROME&362&3e-99
H13	IB0AAA20CH08EM1	20	P15	Churchill protein - Xenopus laevis (African cl...	Q9DFZ4 CHUR_XENLA&125&5e-28
H14	IB0AAA20DF03EM1	20	L6	Forkhead box protein B1 - Xenopus laevis (Afr...	O93529 FOXB1_XENLA&235&3e-61
H15	IB0AAA21CF12EM1	21	L23	Class B basic helix-loop-helix protein 8 - Ratt...	P70562 BHLH8_RAT&63&2e-09
H16	IB0AAA21DG1FM1	21	N2	Transcription factor Ovo-like 2 - Mus musculu...	Q8CIV7 OVOL2_MOUSE&69&6e-11
H17	IB0AAA22CD03EM1	22	H5	REST corepressor 3 - Homo sapiens (Human)	Q9P2K3 RCOR3_HUMAN&260&2e-75
H18	IB0AAA23CD11EM1	23	H21	PHD finger protein 12 - Mus musculus (Mouse)	Q5SPL2 PHF12_MOUSE&182&4e-45
H19	IB0AAA23CE06EM1	23	J11	Endothelial differentiation-related factor 1 h...	Q5ZMC0 EDF1_CHICK&204&9e-52
H20	IB0AAA25DF03EM1	25	L6	Cell death specification protein 2 - Caenorhab...	Q94126 CES2_CAEEL&89&2e-17
H21	IB0AAA27BE01EM1	27	I2	Pre-B-cell leukemia transcription factor 3 - M...	O35317 PBX3_MOUSE&220&7e-70
H22	IB0AAA27DA08EM1	27	B16	Nuclear transcription factor Y subunit beta - ...	P25210 NFYB_PETMA&159&2e-38
H23	IB0AAA27DB07EM1	27	D14	Mesoderm induction early response protein 1 ...	Q8N108 MIER1_HUMAN&143&4e-69
H24	IB0AAA27DH04EM1	27	P8	Neurogenic differentiation factor 1 - Xenopus ...	Q91616 NDF1_XENLA&157&7e-38
H25	IB0AAA28AG09EM1	28	M17	Thyrotroph embryonic factor - Rattus norvegicus (...)	P41224 TEF_RAT&89&3e-17
H26	IB0AAA28BE11EM1	28	I22	Achaete-scute homolog 1a - Danio rerio (Zebra...	Q90259 ASL1A_DANRE&85&4e-16
H27	IB0AAA28BG11EM1	28	M22	Transcription factor 12 - Homo sapiens (Human)	Q99081 HTF4_HUMAN&125&3e-28
H28	IB0AAA28CE03EM1	28	J5	LIM domain transcription factor LMO4 - Mus mus...	P61969 LMO4_MOUSE&222&2e-57
H29	IB0AAA28CH12EM1	28	P23	Transcription factor SUM-1 - Lytechinus varieg...	Q00492 SUM1_LYTVA&153&1e-36
H30	IB0AAA28DD02EM1	28	H4	Nuclear factor NF-kappa-B p105 subunit - Mus ...	P25799 NFkB1_MOUSE&290&6e-78
H31	IB0AAA29BB12EM1	29	C24	Fos-related antigen 2 - Rattus norvegicus (Rat)	P51145 FOSL2_RAT&52&3e-06
H32	IB0AAA29DB01FM1	29	D21	Homeobox protein PKNX1 - Homo sapiens (Human)	P55347 PKNX1_HUMAN&250&3e-71
H33	IB0AAA30BG06EM1	30	M12	Y-box factor homolog - Aplysia californica (C...	P41824 YBOXH_APLCA&182&2e-45
H34	IB0AAA30CH08EM1	30	P15	Insulin gene enhancer protein isl-1 - Danio re...	P53405 ISL1_DANRE&323&9e-88
H35	IB0AAA31CC01EM1	31	F1	Transcription factor HES-1 - Rattus norvegicus (...)	Q04666 HES1_RAT&65&5e-10
H36	IB0AAA32AG02EM1	32	M3	Nuclear receptor corepressor 1 - Xenopus trop...	O4KKX4 NCOR1_XENTRA&102&2e-21
H37	IB0AAA32BB09EM1	32	C18	Glucocorticoid modulatory element-binding pro...	Q2HJ87 GMEB1_BOVIN&98&5e-20
H38	IB0AAA32BG02EM1	32	M4	Kruppel-like factor 4 - Homo sapiens (Human)	O43474 KLF4_HUMAN&109&3e-32
H39	IB0AAA32DD08EM2	32	H16	DNA-binding protein P3A2 - Strongylocentrotus ...	Q04073 P3A2_STRPU&286&5e-82
H40	IB0AAA33AC08EM1	33	E15	POU domain, class 3, transcription factor 2-A...	P31365 P3F2A_XENLA&270&6e-72
H41	IB0AAA33BF01EM1	33	K2	Tripartite motif-containing protein 56 - Mus ...	Q80V11 TRIM5_MOUSE&138&4e-32
H42	IB0AAA33BF04EM1	33	K8	X box-binding protein 1 - Rattus norvegicus (Rat)	Q9R1S4 XBP1_RAT&117&7e-26
H43	IB0AAA33DC04EM1	33	F8	Pituitary homeobox 2 - Rattus norvegicus (Rat)	Q9R0W1 PITX2_RAT&147&1e-34
H44	IB0AAA33DF04EM1	33	L8	Transcription factor SOX-11 - Mus musculus (M...	Q7M6Y2 SOX11_MOUSE&159&2e-38
H45	IB0AAA34AG01EM1	34	M1	Tumor protein p73-like - Rattus norvegicus (Rat)	Q9JJP6 P73L_RAT&248&3e-65
H46	IB0AAA34AH12EM1	34	O23	Transcription factor AP-1 - Bos taurus (Bovine)	O77627 JUN_BOVIN&121&6e-27
H47	IB0AAA35DD10EM1	35	H20	Forkhead box protein N3 - Homo sapiens (Human)	O00409 FOXP3_HUMAN&236&1e-61
H48	IB0AAA35DF09FM1	35	L18	Cyclic AMP-dependent transcription factor ATF...	Q9Y2D1 ATF5_HUMAN&77&1e-13
H49	IB0AAA37AB06EM1	37	C11	SGT1 protein homolog - Mus musculus (Mouse)	Q9CS74 SGT1_MOUSE&317&4e-86
H50	IB0AAA37AB09EM1	37	C17	Methyl-CpG-binding domain protein 3 - Homo sap...	O95983 MBD3_HUMAN&256&9e-68
H51	IB0AAA37AF01EM1	37	K1	Hepatocyte nuclear factor 3-gamma - Bos tauru...	Q3Y598 HNF3G_BOVIN&101&6e-21
H52	IB0AAA37BD02EM1	37	G4	Nuclear hormone receptor family member nhr-23...	P41828 NHR23_CAEEL&106&2e-22
H53	IB0AAA40CH12EM1	40	P23	Hairy/enhancer-of-split related with YRPW motif...	Q7KM13 HEY_DROME&186&2e-46
H54	IB0AAA40DF05EM1	40	L10	Diencephalon/mesencephalon homeobox protein 1...	O566X8 DMX1B_DANRE&129&4e-29
H55	IB0AAA41BB12EM1	41	C24	Uncharacterized potential DNA-binding protein...	Q9DCT6 CQ049_MOUSE&105&4e-22
H56	IB0AAD10YG23CM1	10	G23	Sterol regulatory element-binding protein 2 - ...	Q12772 SRBP2_HUMAN&124&4e-28
H57	IB0AAD10YJ04CM1	10	J4	Myc-associated zinc finger protein - Mus muscul...	P56671 MAZ_MOUSE&68&5e-11
H58	IB0AAD11YG06CM1	11	G6	Myb-related protein A - Gallus gallus (Chicken)	P52550 MYBA_CHICK&169&1e-49
H59	IB0AAD14YN03CM1	14	N3	Negative cofactor 2-beta - Drosophila melanoga...	Q9VJQ5 NC2B_DROME&72&5e-12
H60	IB0AAD15YB15CM1	15	B15	Helix-loop-helix protein 6 - Caenorhabditis el...	Q10007 HLH6_CAEEL&57&8e-07
H61	IB0AAD15YB17CM1	15	B17	Transcription factor CP2 - Gallus gallus (Chi...	Q772U9 TFPC2_CHICK&72&4e-12
H62	IB0AAD16YL13CM1	16	L13	Transcriptional adapter 2B - Drosophila melan...	Q8I8V0 ADA2B_DROME&115&3e-25
H63	IB0AAD19YV11CM1	19	C11	Transcriptional repressor protein YY1 - Mus mu...	Q00899 YY1_MOUSE&100&1e-20
H64	IB0AAD25YD13CM1	25	D13	Transcription factor Maf - Rattus norvegicus (Rat)	P54844 MAF_RAT&75&9e-13

H65	IB0AAD26YK15CM1	26	K15	Thyroid hormone receptor alpha - Rana catesbeia...	Q02777 THA_RANCA&152&4e-36
H66	IB0AAD32YE09CM1	32	E9	Protein krueppel - Drosophila melanogaster (Fr...	P07247 KRUP_DROME&84&1e-15
H67	IB0AAD32YF21CM1	32	F21	DNA-binding protein Ewg - Drosophila melanogast...	Q24312 EWG_DROME&54&2e-06
H68	IB0AAD34YK04CM1	34	K4	Zinc finger protein OZF - Pongo pygmaeus (Orang...	Q5RFP4 OZF_PONPY&114&7e-25
H69	IB0AAD36YP21CM1	36	P21	Hepatic leukemia factor - Homo sapiens (Human)	Q16534 HLF_HUMAN&137&1e-31
H70	IB0AAD4YE19CM1	4	E19	Chromodomain helicase-DNA-binding protein Mi-2...	O97159 CHDM_DROME&144&3e-34
H71	IB0AAD5YG04CM1	5	G4	DNA-binding protein D-ETS-3 - Drosophila melan...	P29774 ETS3_DROME&262&1e-69
H72	IB0AAD5YJ03CM1	5	J3	GATA zinc finger domain-containing protein 1 ...	Q1L8G7 GATD1_DANRE&181&2e-46
H73	IB0AAD6YC18CM1	6	C18	Alpha-fetoprotein enhancer-binding protein - ...	Q15911 ATBF1_HUMAN&52&3e-06
H74	IB0AAD7YM06CM1	7	M6	Transcriptional repressor NF-X1 - Homo sapiens...	Q12986 NFX1_HUMAN&88&5e-17
H75	IB0AAD7YM19CM1	7	M19	Transcriptional regulator ATRX - Mus musculus ...	Q61687 ATRX_MOUSE&155&2e-37
H76	Pdu_48_2_E02			Protein deadpan - Drosophila melanogaster (Frui...	Q26263 DPN_DROME&59&2e-08
H77	Pdu_48_5_E12			Forkhead box protein K2 - Mus musculus (Mouse)	Q3UCQ1 FOXK2_MOUSE&148&2e-35
H78	IB0AAA15AC10EM1	15	E19	PR domain zinc finger protein 10 - Homo sapie...	Q9NQV6 PRD10_HUMAN&171&5e-42
H79	IB0AAA15AG12EM1	15	M23	Transcription factor E2F8 - Homo sapiens (Human)	A0AVK6 E2F8_HUMAN&203&8e-59
H80	IB0AAA17CF09EM1	17	L17	Zinc finger protein 182 - Homo sapiens (Human)	P17025 ZN182_HUMAN&158&4e-38
H81	IB0AAA18CA01EM1	18	B1	Transcription factor E2F4 - Homo sapiens (Human)	Q16254 E2F4_HUMAN&288&4e-77
H82	IB0AAA18DA01EM1	18	B2	Zinc finger MYM-type protein 4 - Mus musculus...	A2A791 ZMYM4_MOUSE&321&6e-07
H83	IB0AAA19BA04EM1	19	A8	Zinc finger protein 207 - Homo sapiens (Human)	Q43670 ZN207_HUMAN&144&1e-33
H84	IB0AAA19AF04EM1	19	K7	Gastrula zinc finger protein xFG20-1 - Xenopus...	P18714 ZG20_XENLA&69&5e-11
H85	IB0AAA20BB09EM1	20	C18	ETS translocation variant 5 - Mus musculus (Mo...	Q9CXC9 ETV5_MOUSE&69&5e-11
H86	IB0AAA20BB09FM1	20	C18	ETS translocation variant 1 - Mus musculus (Mo...	P41164 ETV1_MOUSE&221&6e-57
H87	IB0AAA22BD10FM1	22	G20	Chromodomain-helicase-DNA-binding protein 5 - ...	Q8TDI0 CHD5_HUMAN&244&8e-64
H88	IB0AAA23CA06EM1	23	B11	Zinc finger protein 311 - Homo sapiens (Human)	Q5JNZ3 ZN311_HUMAN&66&8e-21
H89	IB0AAA28AC05FM1	28	E9	High mobility group protein 2-like 1 - Homo s...	Q9UGU5 HM2L1_HUMAN&92&7e-18
H90	IB0AAA29BE08FM1	29	I16	Mothers against decapentaplegic homolog 4 - M...	P97471 SMAD4_MOUSE&256&2e-67
H91	IB0AAA32CE06EM1	32	J11	Zinc finger protein 318 - Homo sapiens (Human)	Q5VUA4 ZN318_HUMAN&169&2e-41
H92	IB0AAA34AC05EM1	34	E9	PR domain zinc finger protein 8 - Mus musculu...	Q8BZ97 PRDM8_MOUSE&90&2e-17
H93	IB0AAA34BD03EM1	34	G6	Zinc finger protein ZIC 1 - Homo sapiens (Human)	Q15915 ZIC1_HUMAN&387&e-107
H94	IB0AAA35DD02EM1	35	H4	Transcription factor ETV6 - Mus musculus (Mouse)	P97360 ETV6_MOUSE&52&5e-06
H95	IB0AAA35BF03EM1	35	K6	High mobility group protein DSP1 - Drosophila ...	Q24537 HMG2_DROME&229&2e-59
H96	IB0AAA35DG07EM1	35	N14	Zinc finger protein 341 - Homo sapiens (Human)	Q9BYN7 ZN341_HUMAN&365&e-100
H97	IB0AAA37DA10EM1	37	B20	Zinc finger protein 782 - Homo sapiens (Human)	Q6ZMW2 ZN782_HUMAN&75&5e-13
H98	IB0AAA39CG01FM1	39	N1	Oocyte zinc finger protein XICOF8.4 - Xenopus ...	P18753 Z084_XENLA&69&6e-11
H99	IB0AAA40CE03EM1	40	J5	Chromodomain-helicase-DNA-binding protein 4 - ...	Q6PDQ2 CHD4_MOUSE&465&e-130
H100	IB0AAA42CB05EM2	42	D9	Zinc finger protein 287 - Pongo pygmaeus (Ora...	A2T812 ZN287_PONPY&173&1e-42
H101	IB0AAA42BC09FM1	42	E18	Chromodomain-helicase-DNA-binding protein 2 - ...	O14647 CHD2_HUMAN&327&7e-89
H102	IB0AAA42AC11EM1	42	E21	Zinc finger protein 649 - Homo sapiens (Human)	Q9BS31 ZN649_HUMAN&151&6e-36
H103	IB0AAD6YJ22CM1	6	J22	PR domain zinc finger protein 16 - Homo sapie...	Q9HAZ2 PRD16_HUMAN&74&9e-13
H104	IB0AAD9YM23CM1	9	M23	Zinc finger protein 605 - Homo sapiens (Human)	Q86T29 ZN605_HUMAN&173&6e-43
H105	IB0AAD11YF21CM1	11	F21	Zinc finger protein 540 - Pongo pygmaeus (Ora...	Q5R5S6 ZN540_PONPY&100&2e-20
H106	IB0AAD16YH01CM1	16	H1	AN1-type zinc finger protein 6 - Rattus norvegi...	Q6DGF4 ZFN6_RAT&49&1e-06
H107	IB0AAD17YG24CM1	17	G24	Zinc finger protein 277 - Homo sapiens (Human)	Q9NRM2 ZN277_HUMAN&186&3e-46
H108	IB0AAD17YK20CM1	17	K20	Zinc finger protein 816A - Homo sapiens (Human)	Q0VGE8 Z816A_HUMAN&82&5e-15
H109	IB0AAD17YL03CM1	17	L3	Zinc finger and BTB domain-containing protein...	Q8NAP8 ZBTB8_HUMAN&71&8e-12
H110	IB0AAD18YC17CM1	18	C17	Chromodomain-helicase-DNA-binding protein 7 - ...	A2AJK6 CHD7_MOUSE&205&2e-52
H111	IB0AAD19YG23CM1	19	G23	Zinc finger protein 664 - Mus musculus (Mouse)	Q4VA44 ZN664_MOUSE&67&2e-10
H112	IB0AAD19YP03CM1	19	P3	Zinc finger protein 583 - Homo sapiens (Human)	Q96ND8 ZN583_HUMAN&75&7e-13
H113	IB0AAD22YI07CM1	22	I7	Mothers against decapentaplegic homolog 3 - Rat...	P84025 SMAD3_RAT&122&3e-27
H114	IB0AAD23YP24CM1	23	P24	High mobility group protein 20A - Mus musculu...	Q9DC33 HM20A_MOUSE&168&4e-41
H115	IB0AAD26YK11CM1	26	K11	Zinc finger protein 84 - Homo sapiens (Human)	P51523 ZNF84_HUMAN&149&4e-35
H116	IB0AAD29YB13CM1	29	B13	Zinc finger protein 509 - Mus musculus (Mouse)	Q8BXX2 ZN509_MOUSE&61&1e-08
H117	IB0AAD29YC15CM1	29	C15	Zinc finger protein 585A - Pongo pygmaeus (Or...	Q5RDX1 Z585A_PONPY&125&9e-30
H118	IB0AAD36YK06CM1	36	K6	Oocyte zinc finger protein XICOF6 - Xenopus lae...	P18749 Z06_XENLA&213&1e-54

