UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE FÍSICA

Unsupervised behavioral classification with 3D pose data from tethered *Drosophila Melanogaster*

João Henrique Fróis Lameiras Campagnolo

Mestrado em Engenharia Biomédica e Biofísica
Perfil em Biofísica Médica e Fisiologia de Sistemas

Versão Provisória

Dissertação orientada por:
Dr. Hugo Alexandre Ferreira
Dr. Pavan Ramdya

2020

Acknowledgements

To everyone that helped and supported me throughout this trying, but rewarding journey, I would like to express my sincere gratitude. First and foremost, I would like to thank Dr. Pavan Ramdya, for welcoming me to his wonderful team and providing me with guidance and confidence to accomplish this endeavor. In addition, I would like to thank my laboratory supervisor, Semih Günel, for always being patient and available to help, but mostly for his friendliness. Likewise, a huge thank you to every member of *Ramdya Lab*, of which I am proud to have been a part of - I hope I was able to retribute your trust with my work during my time with you.

My time in Switzerland could not have been so accommodating if not for the intervention of Miguel and Patricia Gomes, and their lovely sons, Kika and Vicente, who made me a part of their family throughout my time there.

Furthermore, I would like to thank my friends, for all their company and support. To Rodrigo, for all the tennis matches, the discussions, and whose presence I value highly. To Tiago, for his unique kindness and attention. To Francisco, for all the laughs and for all the adventures he got me to be a part of. To Rita, for her unpredictable and whimsical takes, from which I learn a great deal. To Gui, for his antics. To everyone that was a part of my personal odyssey, thank you!

Last, but not the least, I must thank my family for their unconditional support and patience. Writing this essay, by myself, under strenuous personal circumstances posed some afflictions to you as well, but, nonetheless, I always felt the comfort of your presence. Above all, I would like to thank my father, for his sacrifices and for backing my decisions; my mother, for her warm encouragement; my uncle, for his tutoring and guidance; Camille, for her valuable support; and, in particular, my grandmother, for always having faith in me and for her perseverance.

Abstract

One of the preeminent challenges of Behavioral Neuroscience is the understanding of how the brain works and how it ultimately commands an animal's behavior. Solving this brain-behavior linkage requires, on one end, precise, meaningful and coherent techniques for measuring behavior. Rapid technical developments in tools for collecting and analyzing behavioral data, paired with the immaturity of current approaches, motivate an ongoing search for systematic, unbiased behavioral classification techniques.

To accomplish such a classification, this study employs a state-of-the-art tool for tracking 3D pose of tethered *Drosophila*, *DeepFly3D*, to collect a dataset of *x*-, *y*- and *z*- landmark positions over time, from tethered *Drosophila Melanogaster* moving over an air-suspended ball. This is succeeded by unprecedented normalization across individual flies by computing the angles between adjoining landmarks, followed by standard wavelet analysis. Subsequently, six unsupervised behavior classification techniques are compared - four of which follow proven formulas, while the remaining two are experimental. Lastly, their performances are evaluated via meaningful metric scores along with cluster video assessment, as to ensure a fully unbiased cycle - from the conjecturing of a definition of behavior to the corroboration of the results that stem from its assumptions.

Performances from different techniques varied significantly. Techniques that perform clustering in embedded low- (two-) dimensional spaces struggled with their heterogeneous and anisotropic nature. High-dimensional clustering techniques revealed that these properties emerged from the original high-dimensional posture-dynamics spaces. Nonetheless, high and low-dimensional spaces disagree on the arrangement of their elements, with embedded data points showing hierarchical organization, which was lacking prior to their embedding. Low-dimensional clustering techniques were globally a better match against these spatial features and yielded more suitable results. Their candidate embedding algorithms alone were capable of revealing dissimilarities in preferred behaviors among contrasting genotypes of *Drosophila*. Lastly, the top-ranking classification technique produced satisfactory behavioral cluster videos (despite the irregular allocation of rest labels) in a consistent and repeatable manner, while requiring a marginal number of hand tuned parameters.

Keywords: *Drosophila*, behavior, posture-dynamics space, unsupervised learning, clustering, t-SNE, PCA, Gaussian Mixture Model, HDBSCAN.

Resumo

Um dos desafios preeminentes no domínio das Neurociências Comportamentais passa por compreender como funciona o cérebro e como este procede na coordenação dos comportamentos dos animais. Para desvendar a correspondência entre cérebro e comportamento são necessárias ferramentas que consigam medir um comportamento de forma precisa, apreciável e coerente. O rápido desenvolvimento de ferramentas que procuram tanto recolher como processar dados comportamentais, aliado à imaturidade das abordagens contemporâneas, motiva uma procura constante de novas técnicas que consigam alcançar uma classificação de comportamentos não supervisionada (logo, imparcial) de forma consistente.

Para alcançar uma tal classificação, este estudo recorre a uma ferramenta recém desenvolvida que regista a pose tridimensional de *Drosophila* fixas, o *DeepFly3D*, para construir um dataset com as coordenadas *x*-, *y*- e *z*-, ao longo do tempo, de posições de referência de *Drosophila Melanogaster* assentes sobre uma bola flutuante. Sucede-se uma operação inovadora de normalização entre indivíduos de diferentes morfologias, que consiste no cálculo de ângulos entre pontos de referência adjacentes, seguida de uma análise de frequência com wavelets. Subsequentemente, procede-se à comparação de seis técnicas de classificação não supervisionadas - quatro das quais replicam técnicas já algo consolidadas e duas das quais são experimentais. Por fim, as suas prestações são avaliadas mediante métricas expressivas, e inspeção visual dos clusters, de forma a consagrar autonomia ao ciclo que começa com a conceção de uma definição de comportamento e acaba com a ratificação dos resultados que resultam das suas conjeturas.

Os desempenhos destas técnicas foram divergentes. As técnicas que implicam clustering em espaços de baixas (duas) dimensões encontraram obstáculos na natureza heterogénea e anisotrópica destes espaços. Por sua vez, as técnicas que implicam clustering em espaços dinâmico-posturais, de altas dimensões, revelam que os mesmos partilham estas propriedades. No entanto, espaços de baixas e altas dimensões encontram-se em desacordo face à organização dos seus elementos - dados projetados em duas dimensões revelaram uma estrutura hierárquica, que não era evidente nos espaços dinâmico-posturais, previamente à sua projeção. Globalmente, técnicas com clustering em baixas dimensões alcançaram melhores resultados face a estas características espaciais. Por si só, os algoritmos candidatos à projeção de dados do espaço dinâmico-postural foram capazes de revelar dissimilitudes na frequência de comportamentos observados por diferentes genótipos de *Drosophila*. Por fim, a técnica de classificação mais capacitada foi capaz de produzir vídeos satisfatórios de clusters comportamentais (apesar da recorrente classificação errónea de alguns segmentos de pausa), de forma consistente e recorrendo a escassos parâmetros de ajuste manual.

Palavras chave: *Drosophila*, comportamento, espaço dinâmico-postural, aprendizagem não supervisionada, clustering, t-SNE, PCA, Gaussian Mixture Model, HDBSCAN.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Precisely understanding the neural mechanisms that govern animal behavior is one of the more captivating challenges of contemporary science. Animal brains process sensory information to regulate key functions, such as perception, motor control, sleep, homeostasis, learning and memory, which ultimately dictate a final output: behavior. Correspondingly, learning about the brain requires doing so in a context of behavior, which calls for techniques that can provide accurate and quantitative descriptions of behavior.

The discipline that is concerned with the objective study of animal behavior is called ethology (Brown & de Bivort, 2018). Early twenty century ethologists used to guide characterizations of behavior by their experience and intuition to pinpoint behaviors that, for the most part, they already expected to see, such as feeding, fighting or mating. With the emergence of upgraded tools for capturing analyzing behavior, the tendency gradually shifted towards quantitative descriptions of behavior, rather than human annotated labels. These quantitative measurements allowed ethologists to find correlations between behavior and neural activity. However, technological advances in behavior measuring techniques have fallen behind those by new methods for measuring and manipulating neural circuitry, such as *optogenetics* (stimulation strategies based on the expression of light sensitive proteins that regulate the electric state of neurons, (Kim et al., 2017)), *connectomics* (comprehensive study of the neural connections in the brain, (van den Heuvel & Sporns, 2019)) and *optical imaging* of neural activity (Wu et al., 2013). To find causal links between brain function and behavior, these methods ought to be paired with consistent, accurate, interpretable and scalable descriptions of behavior (Berman, 2018).

Behavior can be thought of as changes in pose (posture of an animal in three-dimensional space) over time. Quantifying behavior involves recording a direct representation of the animal, typically by means of video data (although accelerometers, reflectors, audio signals are also adopted), from which more abstruse representations are derived. The advent of cameras with improved spatiotemporal resolution and machine vision algorithms propelled the extraction of skeletonized postural representations using raw pixel luminosities from video frames. This granted ethologists the possibility of creating computationally inexpensive and precise postural datasets with high throughput.

The availability of these rich postural datasets propelled the development of new software modules for quantifying behavior, namely, tracking (computing trajectories of one or multiple animals, (Pérez-Escudero et al., 2014)), action classification (detecting specific patterns of action, usually by means of supervised (Dankert et al., 2009) or unsupervised classifiers (Vogelstein et al., 2014)) and behavior analysis (putting together sequential actions to identify large-scale behavioral patterns and possibly uncover underlying decision mechanisms based on the animal's internal state and external stimuli (Luo et al., 2010)). The scope of this thesis falls on the second category of modules: action classification. Historically, ethologists would often quantify an action according to the output of their experimental apparatus (Shoji et al., 2012), from which consistent results with high throughputs would come at a cost of low-dimensionality and unnatural contextual significance. Presently, action classification approaches are categorized as either supervised or unsupervised. Supervised approaches require a manually annotated training set to be fed to a classifier, which is trained to assign the training set's labels to new behavioral instances, according to a learning algorithm of choice. However, reliance on human observation poses some limitations to supervised methods: 1) frame-by-frame labeling is a slow, mind-numbing task; 2) it's subjective, due to its dependence on each observer's experiences; 3) it's low-dimensional (disregards finer components that make up more complex, recognizable actions); 4) it infers that behavior transitions from one discrete state to another, without first deriving this assumption from the data itself; 5) it neglects behaviors that lie outside the researcher's repertoire. Contrastingly, unsupervised methods explore variations among axes of postural or posture-dynamics data, or the natural clusters that it may encompass, while considering as few a priori assumptions as possible. Methods that focus on unsupervised clustering of time series data (including pose data) usually follow a routine formula: 1) Machine vision is employed to capture pose elements in several dimensions; 2) Data preprocessing, including a crucial time-frequency analysis step to grasp posture-dynamics information; 3) Dimensionality reduction is performed over the posture-dynamics' expression matrix to ease computational requirements of subsequent steps and to facilitate interpretation of results (when the data is embedded onto two or three dimensions); 4) Cluster assignment, to provide individual data frames with a label from a discrete list of behavioral modes. Some clustering algorithms require an intermediate density estimation step to derive an estimation of the embedded data's underlying probability density function. Theoretically, some clustering algorithms can bypass step 3) and assign cluster labels directly in the high-dimensional posture-dynamics space. However, this is disregarded, largely due to performance constraints and the effects of the curse of dimensionality (volume increases rapidly with the increase of the number of dimensions, rendering the data excessively sparse, which is problematic for methods that require statistical significance). This formula is a prominent feature of some groundbreaking works' methodologies, such as Berman et al., (*2014*), Todd et al., (*2017*) and Vogelstein et al., (*2014*). However, the performance of unsupervised approaches lacks benchmarks for success due to the scarcity of the assumptions they impose. This motivates manual, qualitative assessments of their results, which challenges the concept of an unbiased, non-subjective approach once again.

This work sets out to find a methodology that can accomplish unsupervised action classification from a dataset of *Drosophila Melanogaster* tridimensional landmark coordinates over time, provided by the state-of-the art *DeepFly3D*, by Günel et al., (*2019*). The strategy to reach this goal complies with the four methodological steps described above: machine vision is performed upon videos of *behaving D. Melanogaster* by *DeepFly3D*; a preprocessing module comprises landmark angle extraction and time-frequency analysis steps; a dimensionality reduction module performs either t-SNE, PCA, or a combination of both to embed high-dimensional posture-dynamics data; a clustering module employs either watershed transforms, Gaussian Mixture Models or HDBSCAN to assign cluster labels to each video frame. This allows for a total of ten pipeline combinations (considering that HDBSCAN can perform clustering directly in high-dimensional spaces), of which six are explored in detail. With the exception of the newfangled HDBSCAN, the dimensionality reduction and clustering modules don't introduce novel algorithms while fulfilling their parts - the resulting combinations that stem from their use were already subject of scrutiny by Berman et al., (*2014*) and Todd et al., (*2017*). Nonetheless, the adoption of an unprecedented 3D pose dataset and an angle-based normalization step may provide new features which might challenge, or boost, the success of these proven endeavors.

In addition to truthful clusters, each populated by instances of a same recognizable behavioral pattern, this essay is concerned with ensuring generalization across individual flies and genotypes and automation of methodological choice parameters and success benchmarks, as to remove human bias from the procedure and, consequently, steer it towards the unveiling of new behaviors, that supervised classifiers are bound to overlook. Generalization is supported by the landmark angle extraction step, which computes meaningful angles from adjacent landmarks (mostly joint angles) to dissipate the effects of disparate individual scales and experimental settings. Automation is promoted by the employment of either semi-empirical laws or a posteriori success benchmark algorithms to assess the values of the more critical hand tuned parameters from each of the algorithms that are adopted at each stage of the clustering pipeline.

This thesis is divided in five chapters (including the current one). The second chapter focuses on establishing a theoretical framework that lays the ground for this endeavor. It oversees the convoluted definition of behavior and some fundamental assumptions in its regard, the anatomy of *D. Melanogaster*, a conceptual description of relevant algorithms and a recapitulation of relevant research in the field of behavioral neuroscience. Chapter three is dedicated to this work's methodology, following the chronological steps of its inception: from the recreation of the algorithm by Berman et al., (*2014*), in a *Python* setting, to its adjustment, rendering it suitable for pose data, and, finally, introduction of modular modifications at the embedding and clustering levels. Quantitative metrics for cluster evaluation are also introduced at this stage. The chapter's closing section acknowledges the logistical and hardware aspects of this work. The results and discussion are introduced in the fourth chapter. Its first subsection is concerned with the effectiveness of the angle normalization step, which is ascertained by comparing pipelines that perform with and without its inclusion by means of quantitative metrics for surveying the embedded data points of each fly. This is followed by the examination of each of the six aforementioned combinatorial pipelines' clustering results via the computing of cluster metric scores and visual inspection of cluster videos (link provided in the methods section). Finally, the conclusion is presented in chapter five, summarizing the results and discussion topics, assessing the mistakes that were carried out and proposing further adjustments in the pursuit of optimal behavior classification.

Chapter 2

Theoretical framework

2.1 A notion of animal behavior

Animal behavior is guided by a combination of genetic mechanisms along with environmental and past-experience contributions. Genes are the blueprints to an organism's anatomy and physiology, and therefore, define its characteristics - for example, the genome dictates the likelihood of developing certain illnesses, which consequently impact one's behavior. Likewise, genetics shape the animal's morphology, which in turn can foreordain its roles, such as the case with *Pheidole pallidula* ant colonies, which comprise morphologically distinct major and minor workers - whereas the former act as nest defenders, the later are foragers – genotypically distinguished by the expression of a single gene (*ppfor*) (Lucas & Sokolowski, 2009). Furthermore, the genome can directly mediate a class of behaviors, named innate behaviors: behaviors that often appear in fully functional form when performed for the first time, are displayed by different individuals of the same species, and persist stably under variable conditions, even when the animal is raised in captivity. Innate behaviors are more prominent in contexts where a species' environment varies little from generation to generation, or in communication when unambiguous messages need to be sent and received (Breed & Sanchez, 2010).

On a broader scale, physiological mechanisms are also linked to behavior. The communication between the multiple systems that sustain an animal's activity is achieved through chemical and electrical pathways. If the communication is affected, the systems are affected, and, implicitly, behavior is affected. In human beings, for example, excessive adipose tissue and insulin resistance contribute to physiological changes associated with inflammation, vascularization, and oxidative stress, which are conducive to deficits in executive function and memory. The impaired cognition affects behavioral choices, promoting behaviors that perpetuate the increase of adiposity and metabolic dysfunction, furthering the cognitive decline (Farruggia & Small, 2019).

Furthermore, by interacting with its environment, an animal can acquire new behavioral patterns, through learning mechanisms, such as conditioning (e.g. taste aversions or preferences) and trial-and-error learning. These mechanisms bestow the animal with solutions to rapid transitions of local conditions or large-scale environmental changes, such as the improvement of predatory skills, or, inversely, predator evasion skills, food storing, foraging, among others… A neat example of an acquired behavior is observed in bearded vultures, *Gypaetus Barbatus*, which learn to make use of the earth's gravitational pull to crack bones that would be otherwise too large to swallow and then feed upon the exposed bone marrow.

Learning from practice would not be possible if an animal had not been previously steered by its genes to behave in a certain way. Genetics gave the animal and its ancestors the predisposition for successful matchups against their environmental challenges and, conversely, the animal's involvement with its surroundings influenced the composition of its genome, through natural selection of favorable genotypes that are conducive to suitable behavioral phenotypes. Accordingly, behavior should be regarded as the result of a complex interaction between an animal's genes and surrounding environment, equipping it with genetic-coded behavior instructions as well as flexible mechanisms that allow it to respond to particular, day-to-day intricacies.

2.2 Principles of behavior

The technical refinement of behavioral study methodologies has bolstered the undercovering of conceptual insights on behavior. Among them feature guidelines, or principles, of behavior, which focus on regularities of behavior that may facilitate further exploration of animal behaviors. While some principles of behavior are more encompassing in terms of validity across species and their environments, others are quite unique. Brown & de Bivort, (2018) highlight four principles of behavior: *low postural dimensionality*, *discretization*, *stereotypy* and *hierarchy* (Figure 2.1). The former three makeup fundamental assumptions of this endeavor's methodology, while the possible existence of a hierarchical structure among the behavioral clusters might be an asset (for example, the presence of hierarchy suggests more favorable matchups with hierarchical clustering techniques). It is also crucial to mention that all four behavior principles hold when transitioning from the posture space to the posture-dynamics space (whose points represent pose *and* its change over time) since the methodology comprises a time-frequency analysis step.

2.2.1 Low postural dimensionality

This property arises from biomechanical correlations between animal body parts, which are convenient when deriving condensed, lower-dimensional postural representations of pose data form animals that comprise many postural degrees of freedom. In *Caenorhabditis Elegans* worms, 95% of the adopted postures can be described by a combination of four fundamental postures, named *eigenworms*, that capture the dynamic behavior of the worm's midline and its overall curvature (Stephens et al., 2008). Partially reconstructing equations of motion along the four principal components yields a set of dynamical attractors that correspond to multiple behavioral states, such as forward crawling, reversals or pauses. Analogous methodologies corroborate the low dimensionality of *C. Elegans* pose data and go as far as revealing some overlooked behaviors, such as the delta turn (Broekmans et al., 2016).

Low dimensionality is not an exclusive feature to animals with simple body architectures. It holds as well for higher level vertebrate motor systems, where voluntary actions are composed of simpler modules that occur either simultaneously or sequentially. This is evidenced in more constrained tasks, such as reaching in primates and even in octopuses, that display a quasi-articulated limb structure when fetching for food, despite their flexibility.

2.2.2 Stereotypy and discretization

Animals are theoretically allowed to cover a vast range of movements and yet, typically, only a fraction of this lot is observed (Katsov et al., 2017). Furthermore, within this small portion of the achievable behavioral repertoire, movements can be assumed to be repeatable and predictable, appearing similar enough to previous instances of the same behavior, but discernible enough from instances of different behaviors. This similarity among repeated occurrences of an action is referred to as *stereotypy* and it can be witnessed in overlapping, or near-by, trajectories in embedded posture-dynamics spaces.

This property has proven very useful in unsupervised methods for mapping behavior that forsake the dependency on human-annotated behavior labels, promoting the generalization of their models and the discovery of new stereotyped actions that had been previously dismissed by researchers (Marques et al., 2018). Such methods are said to be *unbiased* since they place the researcher at the beginning (by defining stereotypy) and at the end (interpreting behavioral outputs) of the methodology, rather than in the middle (intuitively defining behaviors). Stereotyped patterns often reveal *discretization* in low dimensional spaces, where non-stereotyped patterns fill the gaps between stereotyped regions (Berman et al., *2014*).

In biology, while stereotypy allows animals to keep a well-adapted and efficient set of actions to deal with the majority of contexts that they're presented with, an over reliance on stereotyped, patterned actions can come at a cost of flexibility, which is necessary when the animals' environmental settings shift rapidly. Furthermore, stereotypy is linked with predictability, which may be costly for animals (for example, evading experienced predators).

2.2.3 Hierarchy

Animals engage in a multitude of complex behaviors that result from specific series of distinct motor actions. While looking to understand the rules behind the orchestration of individual actions into sequential behaviors by nervous systems, most frameworks adopt hierarchical models. In fairness, neural architectures that oversee animal behavior are anatomically hierarchical, and hierarchy consists of a handy and familiar organizing principle for human beings. In these hierarchical models, actions are allocated into modules, some less encompassing as for simpler, shorter motions, and some more

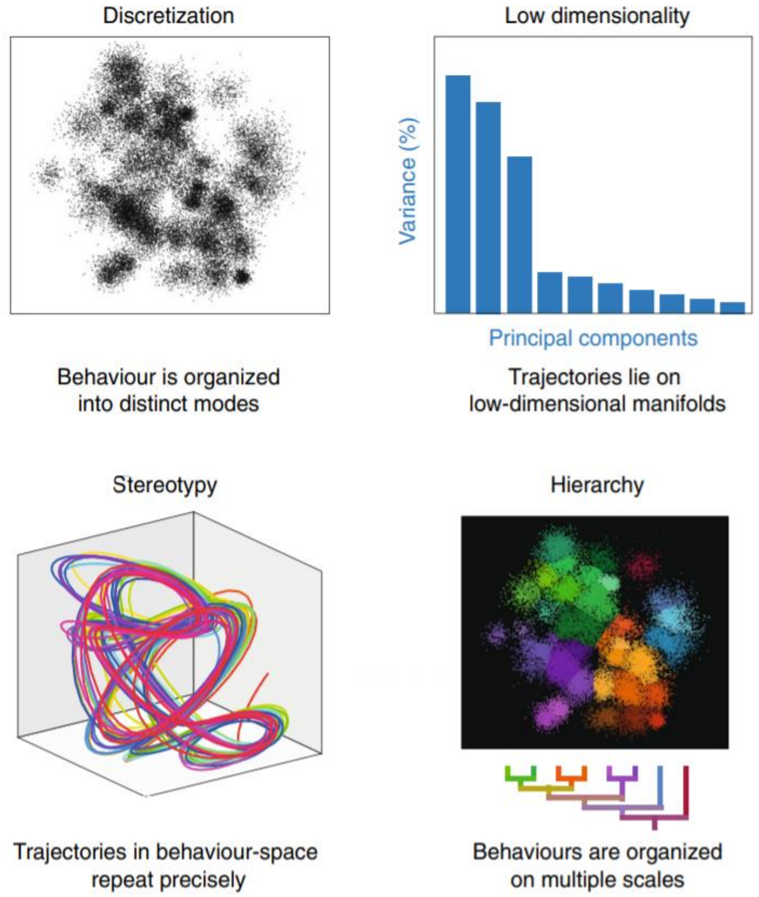


Figure 2.1: Principles of behavior. Behavior often consists of repeatable and distinct movement patterns (*Discretization*) and can be described by a simplified representation of the animal, despite its morphological complexity (*Low dimensionality*). Furthermore, behavior is organized in echelons that are nested into larger, more encompassing echelons (*Hierarchy*). The lower echelons hold highly similar movement patterns (*Stereotypy*). Figure from Brown & de Bivort, (*2018*).

encompassing, as for concatenations of actions. For example, fly motor actions that makeup grooming patterns can be organized according to the particular region of the body they concern (forelegs, eyes, wings…), which are nested into more widespread and complex patterns, involving further actions and specific dynamics (Richard & Dawkins, *1976*).

Hierarchical models can also infer on the genesis of motor function, with two main hypotheses prevailing: 1) in a sequence of actions, each action triggers the following; 2) all the actions are readied in parallel and then selected through a winner-takes-all competition. By generating a computer model that performs hierarchical suppression of motor actions and fitting it to *Drosophila* grooming patterns, Seeds et al., (*2014*) demonstrated that grooming motor actions are organized by a suppression hierarchy in flies.

Inversely, a careless adoption of hierarchical clustering algorithms may result in erroneous presumptions of hierarchy, without this being ratified by the data itself, as a consequence of the algorithms' axioms. Furthermore, such algorithms usually neglect the contribution of multiple time scales, opting for a single scale that is determined by the results of a Markov transition model that only considers transitions from a current state to its successor. This assumption is flawed since, even in simpler animals, both long-term (e.g. hunger) and short-term (e.g. thermotaxis) stimuli help coordinate behavioral transitions.

Although it is not correct to infer that behavior is organized hierarchically from positive results by hierarchical algorithms, hierarchy, as a principle, proves useful in the exploration of behavior (namely due to its success in formulating predictive models). For example, using a *treeness* metric to quantify the degree of hierarchy in *D. Melanogaster*, Berman et al., (*2016*) demonstrated that a nested hierarchical representation could predict longer-scale behaviors beyond the capacities of a Markovian predictive model: fine grained partitions are appropriate in predicting the near future, while coarser partitions are satisfactory for predicting the relatively distant future.

2.3 *Drosophila Melanogaster*

Drosophila Melanogaster is a species of fly, commonly known as fruit fly. Its use in neuroscience, as a model organism, is backed by *D. Melanogaster*'s short life cycle, a large number of offspring per generation and relatively simple and easily manipulable genetics that, when coupled with behavioral measurements, allow researchers to study the genetic grounds of behavior.

Anatomically, the head is the first of three major divisions of *D. Melanogaster*'s body (Figure 2.2). It holds, among other structures, the naturally red compound eyes, the *proboscis* (the mouthparts) and a pair of sensory appendages, the *antennae*. The second major division of the fruit fly's body is the thorax, which is further divided into pro-, meso- and metathorax. The former accommodates the forelegs, while the mesothorax holds the wing muscles and the middlegs, and, finally, the metathorax is connected to the hindlegs. Lastly, the abdomen is the third major division of the insect body and is shielded by the *tergites* on the dorsal side. *Tergites* become progressively darker along the cranial-caudal axis, which gives rise to stripe-like patterns on the fly's abdomen (these stripes are among the landmarks captured by *DeepFly3D*, by Günel et al., (*2019*)). Each *D. Melanogaster* limb is divided into 10 segments: *coxa*, *trochanter*, *femur*, *tibia*, *tarsus* (composed of 5 sub-segments) and *pretarsus* (listed from proximal to distal). The prefixes pro-, meso- and meta- are included when distinguishing between front, middle or hinder leg segments. Relative to females, male *D. Melanogaster* are slightly smaller and present darker backs. In addition, males possess a row of dark bristles on the *protarsus*' first segment, named *sexcombs*, which are used during mating behaviors.

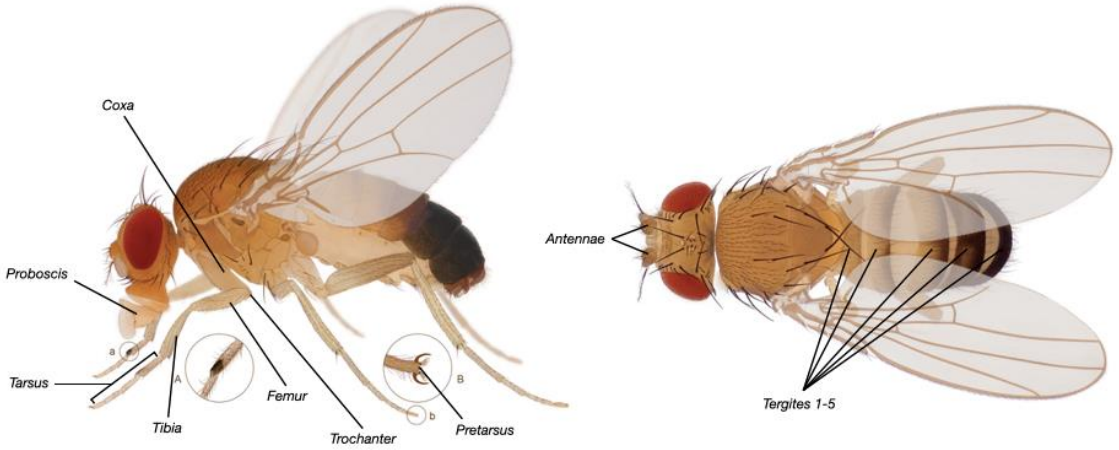


Figure 2.4: Anatomy of *Drosophila Melanogaster*. The images were taken from the *Anatomical Atlas of the male Drosophila melanogaster*, and the relevant labels added manually.

2.4 Exploration of unsupervised behavior classification approaches

The clarification of underlying principles of behavior, such as low dimensionality and stereotypy, has steered much of the recent progress in developing tools for data-driven and unsupervised analysis of animal behavior. Even though the term “unbiased” is often employed, none of the mentioned approaches can be considered strictly unbiased due to the fact they comprise many methodological choice points and tuning parameters. Ethologists are challenged to navigate through them without reverting to a supervised analysis.

Approaches for identifying stereotyped behaviors (including that of this work) share a general framework: they begin by extracting a low-dimensional representation of the animal, the posture space, then followed by the translation of the posture space into a posture-dynamics space, which accounts for the dynamical aspects of behavior; lastly, this dynamical representation is used to generate a behavioral representation whereupon individual stereotyped actions are isolated.

2.4.1 Extracting posture

Posture tracking analyses usually initiate with the recording of raw video data and subsequent isolation of posture from the image background. Here, *posture* should be considered as a descriptive measure of an animal’s anatomy at a given point in time, so that behavior can be measured independently of background or spatial orientation. The aim is to simplify a high dimensional measurement, such as a frame composed of thousands of pixel values, into a low dimensional set of numbers describing the animal’s posture. Different animals require quite specific representations - while, for worms, a centerline representation can account for more than 95% of the variation in naturally-occurring body postures (Broekmans et al., 2016), a fly’s movements are usually best described by the combination of its joint, wing and abdominal coordinates.

Historically, posture tracking assignments demanded mind-numbing manual annotations of video frames, which rendered virtually impossible the creation of large pose datasets. Eventually, tracking markers inspired by human motion capturing techniques were employed by placing contrasting spots (reflective, colored or fluorescent) on the animal's limbs or appendages (Figure 2.3(a)). These could consequently be isolated from the image background and automatically tracked (Bender et al., *2010*). However, these techniques are not well suited for smaller animals, such as D. Melanogaster, since their size makes it difficult to precisely mark small points of interest. Moreover, the markers may hamper the animal's movements and still provide an insufficient 3D kinematic description of its limbs.

In parallel, *marker-less* techniques emerged. Some exploit contact points between an animal and its substrate using optical phenomena (Mendes et al., *2013*), though bounded by solely tracking sections of the animal that make contact with the substrate (Figure 2.3(b)). Instead, computer vision techniques can distinguish raw pixel luminosities from frames aligned with a given orientation of the animal. In such cases, postural representation of the animal is achieved through dimensionality reduction operations. The groundbreaking work of Berman et al., (*2014*) (which will be exhaustively revisited) exploits *Principal Component Analysis* (PCA) to translate the pixel luminosities from segmented, rescaled and aligned frames into a low-dimensional set of 50 time series (termed *postural modes*), explaining roughly 93% of the observed variation. Suchlike tools prove to be versatile in granting reproducible and continuous mappings of behavior from animals with diversified shapes and limb configurations. Yet, they are limited to 2D pose, and fall short when faced with long data sequences, cluttered backgrounds, fast motion and naturally occurring occlusions, when only a single 2D perspective of the animal is offered. Furthermore, they lack interpretability, which should be considered when aiming at a meaningful dynamical representation of behavior.

More recent methods make use of deep learning or neural networks to replicate tracking tasks accurately and automatically, while imposing less sensory or motor constraints to the animal. Their success is boosted by the availability and richness of large sets of annotated data to train deep networks effectively. This has been evidenced by monocular 3D human pose estimation algorithms, which now offer remarkable real time results in uncontrolled environments (Martinez et al., *2017*). Shortcomings with monocular approaches due to occlusions can be solved by using multi-camera setups and triangulating 2D detections, improving accuracy and warranting the elimination of false detections. 2D pose estimation techniques already show auspicious results in laboratory settings, as demonstrated by open-source algorithms such as *DeepLabCut* (based on a state-of-the-art human pose estimation network, *DeeperCut*) (Mathis et al., *2018*) and *LEAP* (Pereira et al., *2019*), portrayed in Figure 2.3(c).

Nonetheless, due to the importance of perspective in behavior quantification and the predisposition to loss of information from occlusions, 2D posture tracking techniques are still incapable of providing a complete representation of an animal's behavior. Some techniques allow 2D to 3D translation through the use of calibration boards (Nath et al., *2019*), yet are exclusively suitable to humans and larger animals, overlooking smaller, D. Melanogaster-sized animals. To include them would require the fabrication of a prohibitively small checkerboard pattern, and a morose process of using a small, external calibration pattern. Flies, in particular, pose further challenges due to being translucent, baring multiple appendages and joints, and requiring the use of infrared light to avoid visual stimulation.

Recently, Günel et al., (*2019*) introduced *DeepFly3D*, a deep learning-based software pipeline that achieves rapid, reproducible and comprehensive 3D pose estimation in tethered adult *D. Melanogaster* (Figure 2.3(d)). It takes in synchronized video data from a multi-camera setup, feeds it to a state-of-the-art deep network (Newell et al., *2016*) and then enforces consistency across views. This allows the network to eliminate spurious detections and use the detected 3D pose errors to further calibrate the deep network, contributing to higher 3D accuracy. Furthermore, *DeepFly3D* optimizes the calibration process by taking the fly *itself* as the calibration target and employing sparse bundle adjustment methods, relieving the researcher from manufacturing a prohibitively small calibration pattern. Results from unsupervised behavioral embedding of 3D joint and landmark angle data gathered

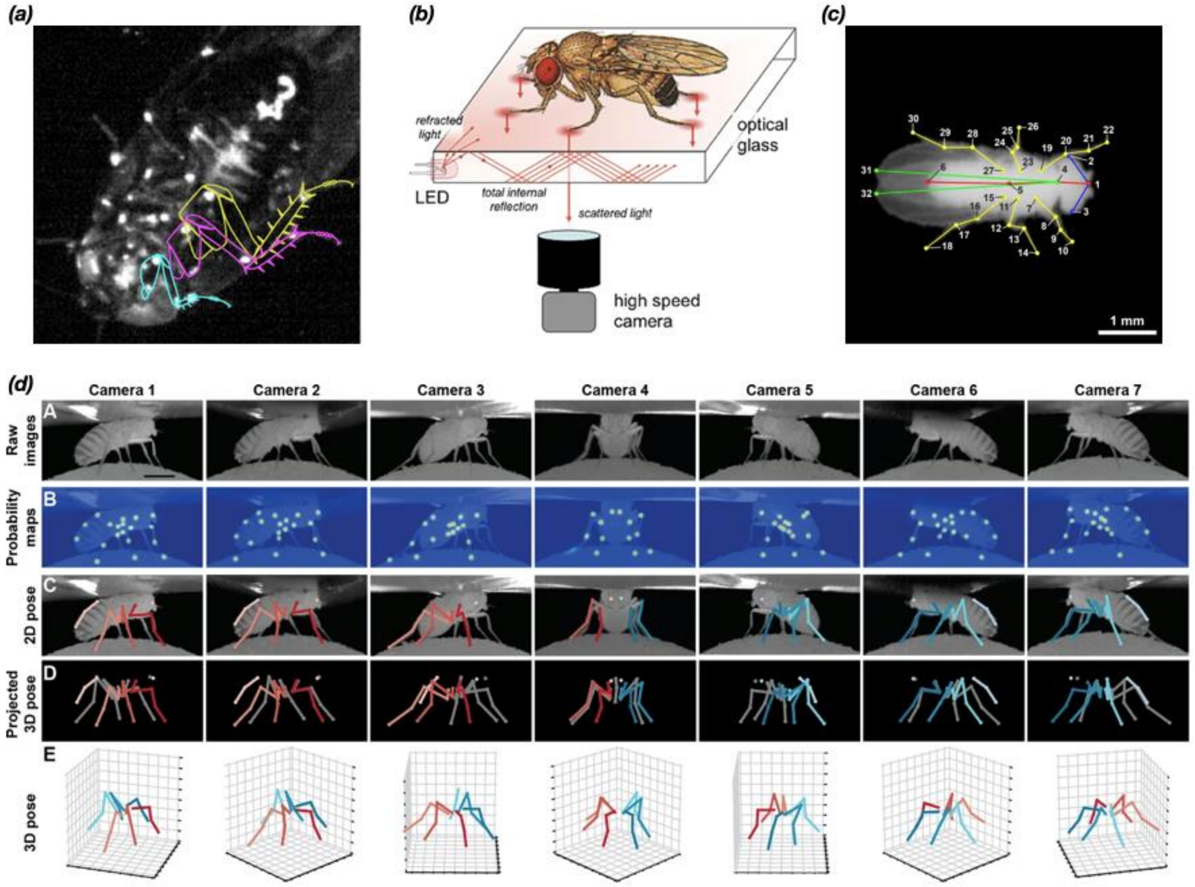


Figure 2.5: Creating a postural representation. **(a)** Motion capturing technique with small dots of reflecting paint placed in the limbs' joints (adapted from (Bender et al., 2010)). **(b)** Description of locomotion parameters with optical touch sensors and high-speed video imaging (adapted from (Mendes et al., 2013)). **(c)** A deep-learning based method for 2D labeling body parts, *LEAP* (adapted from (Pereira et al., 2019)). **(d)** *DeepFly3D*, a deep-learning based software for estimating 3D pose in *D. Melanogaster* (figure from (Günel et al., 2019)).

by *DeepFly3D* prove to be robust in dealing with problematic artifacts that would be noticeable in 2D pose data embeddings. The central endeavor of this thesis is an extension of the work by Günel et al., (2019) - focusing instead in a technique that can derive meaningful representations of behavior from *DeepFly3D*'s pose data.

2.4.1.1 Direct annotation of behaviors from posture

This approach circumvents the construction of a posture-dynamics space altogether and instead opts to directly annotate behaviors from a set of manually curated features, by means of supervised classifiers or clustering, embedding algorithms. Supervised classifiers ask users to intuitively annotate a small set of video frames that are then fed to a classifier that can automatically label behaviors in screen-scale datasets. Algorithms such as *JAABA* (Kabra et al., 2013) can be fast (classifier training times under 40 seconds) and adjustable to different tracking systems and, therefore, different animals and even species. Nonetheless, users are required to encode their intuitive definitions of behavior to train the classifier, which not only can be mind-numbing, but may also introduce a human bias and dismiss more subtle elements of behavior.

Alternatively, cluster algorithms can independently classify data elements into specific groups, according to their similar, or dissimilar, properties, like spatial distribution or hierarchy, among others. This usually requires computing the similarity between every pair of elements, meaning the runtime increases with the square of the number of elements (noted in complexity notation as $O(N^2)$, where N is the number of elements). Therefore, such algorithms are highly impractical when dealing with large datasets. *K-means* (Pelleg & Moore, 1999) is the more prominent clustering algorithm, possessing linear complexity, $O(N)$, and distinguishes classes according to centroid positions. It starts measuring the distances between every point and the *k* classes' randomly assigned centroids and classifying each point according to its closest centroid. The *k* centroids' positions are updated by averaging the positions of their classes' elements. These steps are iterated until the centroids' positions converge. However, *K-means* requires the user to provide the number of classes, *k*, beforehand. Furthermore, since the centroids are assigned random initial coordinates, the results lack repeatability and consistency. *Density Based Spatial Clustering of Applications with Noise* (DBSCAN) (Ester et al., 1996), in its turn, doesn't require a preset number of classes, soliciting only two parameters: ε specifies the neighborhood of each point, and *minPoints* yields the minimum number of points necessary to compose a cluster. Starting with a random point, it assigns other points within ε to its cluster and then repeats this for each new cluster element, until it cannot find new members. At this stage it moves towards a new point, setting up a new cluster. Isolated points, within ε and *minPoints*, are labeled as noise. DBSCAN offers a good matchup against clusters of different sizes and shapes although it struggles with clusters of varying densities. Besides, in very high dimensional spaces, it is very hard to properly estimate ε. On their hand, unsupervised clustering algorithms employing *Gaussian Mixture Models* (GMM) (Marwala, 2018) assume that the data holds a given number of normally distributed subpopulations, which is a broader assumption than that of *K-means*. Hence, two parameters define the shape of the clusters - their *mean* and *standard deviation*. After the user inputs the number of subpopulations (therefore, clusters to be found), the algorithm starts by randomly selecting the gaussian parameters and assigning each data element the probability of belonging to a particular cluster. Gaussian parameters are then updated through a weighted sum of the data elements' positions, where the weights are the aforementioned probabilities. This process is iterated until convergence occurs. GMMs are more flexible to cluster shapes than *K-means* and allow for mixed-membership of data points, but nonetheless still require the user to estimate the number of clusters *a priori*.

A substitute approach to high dimensional clustering of postural features involves embedding algorithms that aim to simplify high-dimensional data, lowering the number of dimensions, while minimizing the information lost in doing so. They operate under the assumption that data elements lie on an embedded manifold within the higher dimensional space. Unlike clustering algorithms, these algorithms allow researchers to visualize the data in the embedded low-dimensional space, rendering it interpretable. Among a multitude of candidates, this essay explores two prominent, but contrasting dimensionality reduction algorithms: *Principal Component Analysis* (PCA) and *t-distributed Stochastic Neighbor Embedding* (t-SNE). PCA (Jollife & Cadima, 2016) prioritizes the preservation of global space structure (i.e. statistical information), while sacrificing local verisimilitude. It does so by creating new uncorrelated variables, the *principal components,* while maximizing the variance in the data observation-variable matrix, noted as $M_{ov.}$ This involves taking the eigenvectors and (orthogonal) eigenvalues of the covariance matrix of M_{ov} and sorting them according to the magnitude of the eigenvalues. Then, the user is left to decide how many principal components (eigenvectors) to keep: 2 or 3 are useful for visualization purposes, or, when not concerned with visualization, the user can calculate the proportion of variance explained for each feature and select a threshold, keeping principal components until the threshold is met. PCA yields high accuracies at quick computing speeds, although it is not particularly skilled at preserving local data structures. To find stereotyped behaviors, for once, embedding should minimize any local distortions, without necessarily attempting the preservation of longer length scales. At this end of the spectrum lies t-SNE (Van Der Maaten & Hinton, 2008), which delivers just that. It starts by measuring similarities between points in the high dimensional space, taking the transition probabilities, noted P_{ij}, between every pair of near neighbors, according to a Gaussian kernel of the user-assigned distance function. The number of nearest neighbors is restricted by the algorithm's *perplexity* parameter, usually ranging between 5 and 50. Then, a new set of transition probabilities, Q_{ij}, is computed, with the particularity that they're now proportional to a Student t-distribution kernel of the points' Euclidean distances in the embedded space. The Student t-distribution kernel bears heavier tails than its Gaussian counterpart, which allows for better modelling of far apart distances. Finally, the algorithm approximates the high- and low-dimensional space transition probabilities, P_{ij} and Q_{ij} respectively, by minimizing a Kullback-Leibler (KL) divergence cost function with *gradient descent*. t-SNE still poses computational complexity, $O(N^2)$, and consistency (the initial iteration is random) issues, but is, nonetheless, the more adequate algorithm to deal with convoluted, non-linear datasets.

Despite the robustness and high throughput of the aforementioned approaches, there is a risk of missing elements of behavioral dynamics due to the reliance on possibly incomplete and biased classifier lists, or dependence on variables that bear different units of measurement (velocities, accelerations, angles…) and therefore require additional conversion factors or assumptions, at risk of altering the outputs in subtle ways. Also, on a footnote, a complete description of all clustering and embedding candidates would be too extensive for this essay. This section is mostly concerned with introducing those that will be relevant to this study's purposes, along with some prominent alternatives.

2.4.2 Characterizing postural dynamics

When looking for a complete, interpretable description of an animal's behavioral repertoire, one focuses on movements, as opposed to isolated postures in time. Hence, the next step typically pursues the conversion of the posture space into a posture-dynamics space, by describing how the postural time series are changing. Thereafter, a behavioral representation can be generated from this dynamical representation of behavior, encompassing longer time scale elements that set up the shorter-scale motions.

Creating a dynamical representation can be accomplished either by directly fitting a differential equation to the postural data or attributing dynamic assimilating features such as temporal motifs or time-frequency analysis features from individual segments of the data. Examples of both approaches are given below.

2.4.2.1 Fitting differential equations to posture time series

2.4.2.1.1 Dynamical systems

An ideal dynamical representation of behavior should emerge naturally from postural dynamics. From human limb movements to animal gait transitions, motor behaviors can be modeled through nonlinear dynamical systems. Stephens et al., (2008) derive a model for the phase dynamics of *C. Elegans* crawling by matching observed trajectories in short timescales with equations of motion that ultimately predict phenomena on much longer timescales (Figure 2.4(a)). Basing their model on the Langevin equation for a Brownian particle subject to forces, using dynamical variables derived from *eigenworms* (low dimensional projections of the worm's posture space) and treating stochastic and deterministic features simultaneously, they find that the worm's behavior can be described by means of a set of dynamic attractors with switching times that can be inferred from the statistics of the stochastic elements (noise).

Uncovering the underlying differential equations directly from the observation of biological systems poses a great challenge to researchers. The structure of these equations is primarily determined either by hand from first principles, regression methods, or nonparametric methods that require assumptions like linearity or the computing of numerical models. However, novel methods (Bongard & Lipson, *2007*; Daniels & Nemenman, *2015*) can automatically uncover the structure and parameters of governing differential equations by iteratively inferring initial conditions, proposing candidate models from the recorded time series of the system's behavior, and generating candidate tests for the competing models. Although directly translating underlying guiding equations from higher dimensional datasets, like those generated by legged animals such as *D. Melanogaster*, remains an onerous task, such progresses provide auspicious pathways for future explorations.

2.4.2.1.2 Statistical models

Ethology proposes that complex behaviors arise from the concatenation of discrete and stereotyped modules of simpler actions. With recent developments in machine vision and machine learning, such behavioral modules and their associated transition probabilities were disclosed in a systematic and reproducible fashion for invertebrate animals. Moreover, these models uncovered context-specific strategies used by the invertebrate brains to adapt behavior during environmental shifts, via the generation of new behavioral modules, or the re-sequencing of current ones. One demonstrative approach was employed by Wiltschko et al., (*2015*), who identified individual modules and assessed their transition probabilities in mice behaving in a synthetic arena (Figure 2.4(b)). After demonstrating from their dataset that mouse dynamics exhibit structure at sub-second time scale, 3D mouse pose data was subjected to wavelet analysis and PCA, and the comprised behavioral modules (less than 1 second in duration), were fitted to different, competing, linear dynamical systems. The best fit to their data was an *Autoregressive Hidden Markov Model* (AR-HMM) that described mouse behavior in a hierarchic manner: the contents of each module reflected the pose dynamics over short time scales, and the transition probabilities stated by the AR-HMM governed the long time scaled relationships between said modules (meaning their order of occurrence).

Despite possessing the ability to simultaneously create dynamical and behavioral representations of behavior (governed by different time scales), this approach is still limited by the necessity of a parameter that sets the overall time scale for individual blocks of behavior. Considering that behavior durations can range from milliseconds (e.g. reflex arcs) to several hours (e.g. sleep), this deficiency can be soothed by adding several time scale parameters when fitting pose data. Nonetheless this solution would still require hand-tuning of the selected time scales, along with a corollary assumption that the time the animal spends in each particular behavior follows an exponential distribution.

2.4.2.2 Multi-scale dynamical representations of posture

2.4.2.2.1 Finding postural motifs

Time series motifs are simply sets of time series, or subsequences of larger time series, that are very similar to each other (Figure 2.4(c)). These hint at an analogous structure that was conserved for a reason of possible interest. For animals, often performing stereotyped behaviors, finding posture-dynamics patterns that commonly occur throughout the dataset consists of an elementary, yet complete, route for characterizing behavior. A popular approach involves reducing the dimensionality of the postural dataset and extract the most repetitive subsequences (behavioral motifs) for each given length, within the limits of the dataset (Brown et al., 2013). Having built a dictionary of motifs for each animal, containing a wide range of behaviors, from single postures to long, complex sequences, it is possible to collect individual “behavioral fingerprints”. These fingerprints reflect the distance between each individual’s behaviors and the dictionary elements, allowing researchers to evaluate phenotypic dissimilarities among individuals, sometimes differing in genotypes, neural manipulations or other conditions of interest. For each animal, a behavioral representation is composed by the ensemble of motifs it performs and their relative frequency and order. Since the considered assumptions in motif extraction were minimal, the method is reproducible throughout different organisms. Nonetheless, it may not always be robust to slight variations between, and within, individuals that result in slight changes in postural dynamics.

2.4.2.2.2 Time-frequency analysis

These techniques take posture time series data and assess the relevant endured frequencies, if they are pertinent to the animal’s behaviors, and output the relative weight of each frequency as a function of time. This task is often accomplished using *Continuous Wavelet Transforms* (CWT) (Griffel & Daubechies, 1995), which make use of inner products to measure the similarity between a signal and an analyzing function. Although similar in principle to the *Short-Time Fourier Transforms* (STFT),

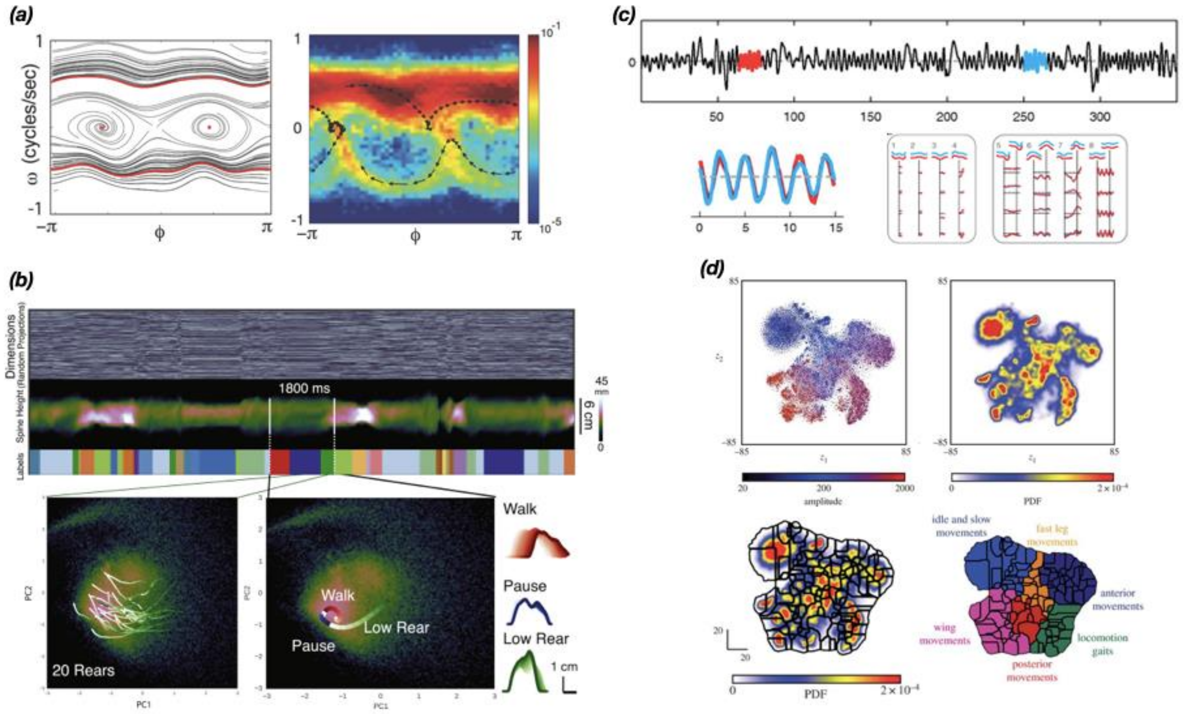


Figure 2.6: Inspecting the dynamics of posture and creating representations of behavior. **(a)** The low dimensional postural structure of *C. elegans* can be parameterized by a single variable, φ, which is fitted to a deterministic dynamical system which yields trajectories that collapse near four attractors, representing well defined classes of behavior (adapted from (Stephens et al., 2008)). **(b)** Projection of mouse spine data onto a *Principal Component* space, which holds templated trajectories, identified by the AR_HMM (adapted from (Wiltschko et al., 2015)). **(c)** Unsupervised detection of behavioral motifs, repetitive patterns in the posture time-series, occurring at a wide range of time scales (adapted from (Brown et al., 2013)). **(d)** t-SNE embedding of posture-dynamics data followed by Gaussian-kernel density estimation and Watershed segmentation yields a 2D behavioral space, partitioned into sectors of stereotyped actions (adapted from (Berman et al., 2014)).

CWT compare the signal to shifted and scaled (compressed or stretched) versions of a wavelet. When employing a complex-valued wavelet, this yields a complex-valued function of two variables: scale and temporal position. Smaller scales indicate that the wavelet is more compressed and is therefore compared to a smaller portion of the signal - rapidly changing details, at higher frequencies, will be measured by the wavelet' coefficients. This provides CWT spectrograms with a multi-resolution time-frequency trade-off property that allows for a more exhaustive description of postural dynamics occurring at different time scales. Equation 3.5 expresses the CWT of a continuous function, $f(t)$, at a scale (s>0), $s \in \mathbb{R}^{+}$, and translational value $\tau \in \mathbb{R}$, where $\psi(t)$ is a continuous function on both time and frequency domains (its overline represents the complex conjugate operation).

$$W\big(s,\tau; f(t), \psi(t)\big) = \frac{1}{|s|^{1/2}} \int_{-\infty}^{\infty} f(t)\overline{\psi}\left(\frac{t-\tau}{s}\right) \qquad 2.1$$

By ignoring phase information and keeping solely the amplitudes of the wavelet coefficients, the need for precise temporal alignment, that is required for motif-based analyses, is suppressed. This culminates in a dynamical representation where, for each postural mode, each point in time is expressed by a set of wavelet magnitudes, each linked with a specific frequency resolution.

2.4.3 Creating a behavioral representation from posture-dynamics spaces

Some of the aforementioned methodologies, namely those that involve supervised behavior annotations and clustering of high-dimensional or embedded posture features (Section 2.4.1.1), directly yield behavioral representations that explain how these posture feature change over time. Posture-dynamics spaces by themselves, however, lack the information necessary to describe longer-time scale changes in the simpler postural motions that make up the coherent, stereotyped patterns associated with behavior (e.g. describing the relative velocities of an insect's limbs *vs* stating that the insect is walking with an alternating tripod gait). Hence, approaches that deal with posture-dynamics spaces require a further step to translate the dynamical representations to behavioral ones, which usually involves the clustering or embedding techniques mentioned above (Figure 2.4(d)). The computed behavioral representations are either discrete (e.g. clusters or motifs) or continuous (e.g. densities or all-including dynamical models). Choosing between them depends on the experimental demands at play: continuous representations allow researchers to capture both stereotyped and non-stereotyped actions (sometimes of equal relevance since animals like flies perform non-stereotyped actions half the time (Berman et al., *2014*)) and allow further discretization without imposing a discrete structure *a priori*; alternatively, discrete representations are a better match when the data does indeed have clusters, since clustering in higher dimensions preserves the integrity of the dataset and is unconcerned with length-scale distortions created by nonlinear embedding algorithms (although it requires overcoming the *curse of dimensionality*). Nonetheless, uncovering both behavioral representations is considered to be a good practice, since this provides additional context and information.

Chapter 3

Materials and methods

3.1 Postural datasets from *DeepFly3D*

As previously mentioned (Section 2.4.1), this research was developed in continuity with *DeepFly3D*'s pipeline (Günel et al., *2019*), which is a deep learning-based software pipeline that achieves rapid, reproducible and comprehensive 3D pose estimation in tethered adult *D. Melanogaster*. *DeepFly3D* is tasked with tracking the coordinates of 4 joints (*body-coxa*, *coxa-femur*, *femur-tibia*, *tibia-tarsus*) plus the *pretarsus* from each of the fly's limbs, 3 abdominal landmarks (*tergite* stripes) on both dextral and sinistral sides of the fly, and 2 antennal landmarks, yielding a total 38 3-dimensional key points per time instance. In the process, it relies on a state-of-the-art *stacked hourglass* human pose estimation network, which was adapted to the experimental input and output constraints. The network's output consists of a stack of probability maps, or *heatmaps*, each encoding the location of one landmark. It tracks 19 2D locations of the same key points, seen across multiple views, for each side of the fly. Rather than selecting random frames for network training purposes, individual camera frames are corrected automatically using frames from other cameras, and persistent errors on multiple camera views are selected for manual annotation and network retraining.

After each 2D landmark is detected for each view, it is still necessary to estimate their matching 3D coordinates. This is achieved through *triangulation* and *bundle-adjustment* to increase the robustness against erroneous detections, while using the fly itself as calibration pattern for the cameras. Since the triangulation procedure might produce erroneous results if the 2D estimates of landmarks are wrong, and considering that each key point is treated independently, there is a need to enforce global geometric constraints. *Pictorial structures* accomplish this by encoding the relationship between a set of variables in a probabilistic setting. Thus, it is possible to consider multiple candidate 2D landmark locations from the network's heatmaps while estimating the 3D coordinates of a landmark point. Triangulating multiple 2D candidates yields a set of 3D candidates for each key point location. The final task consists of finding the most likely one. Solving this from the possible combination of 2D points is NP-hard and, therefore, *DeepFly3D* takes advantage of the non-cyclical graph-like structure of the fly's skeleton to solve the inference problem using *belief propagation*. Lastly, the 3D landmark locations are estimated by selecting the nodes of the graph with the largest belief.

3.2 Implementing a framework for unsupervised behavior quantification in a *Pyhton* setting

To get acquainted with the general framework of unsupervised methods for mapping behavior, an appropriate starting point would be the emulation of the groundbreaking work by Berman et al., (*2014*). This article's pipeline was rebuilt from the ground up in a *Pyhton* environment, as opposed to *MATLAB*, since the former is open-source software, versatile and bares a larger community that can use and perfect the developed code. The original dataset was kindly provided by the author and was employed with the intent of validating the new pipeline's success in between each of its modular steps. Its contents consist of a vast number of frames from individual behaving *D. Melanogaster*, as shown in Figure 3.1. The frames are 200x200 pixel squares and were recorded with a 100Hz high-speed camera, baring sufficient spatiotemporal resolution to detect moving body parts. Due to time and computational memory constraints, only a small portion of the dataset was used.

With the dataset's frames already segmented and aligned, their information is rearranged by means of a Radon transform, and postural decomposition is attained through PCA. The resulting postural modes undergo time-frequency analysis by means of a Morlet wavelet transform, where only the amplitudes of the transform are kept. This yields a high-dimensional space, where each point in time is represented by a vector comprising the amplitudes from each frequency channel within each postural

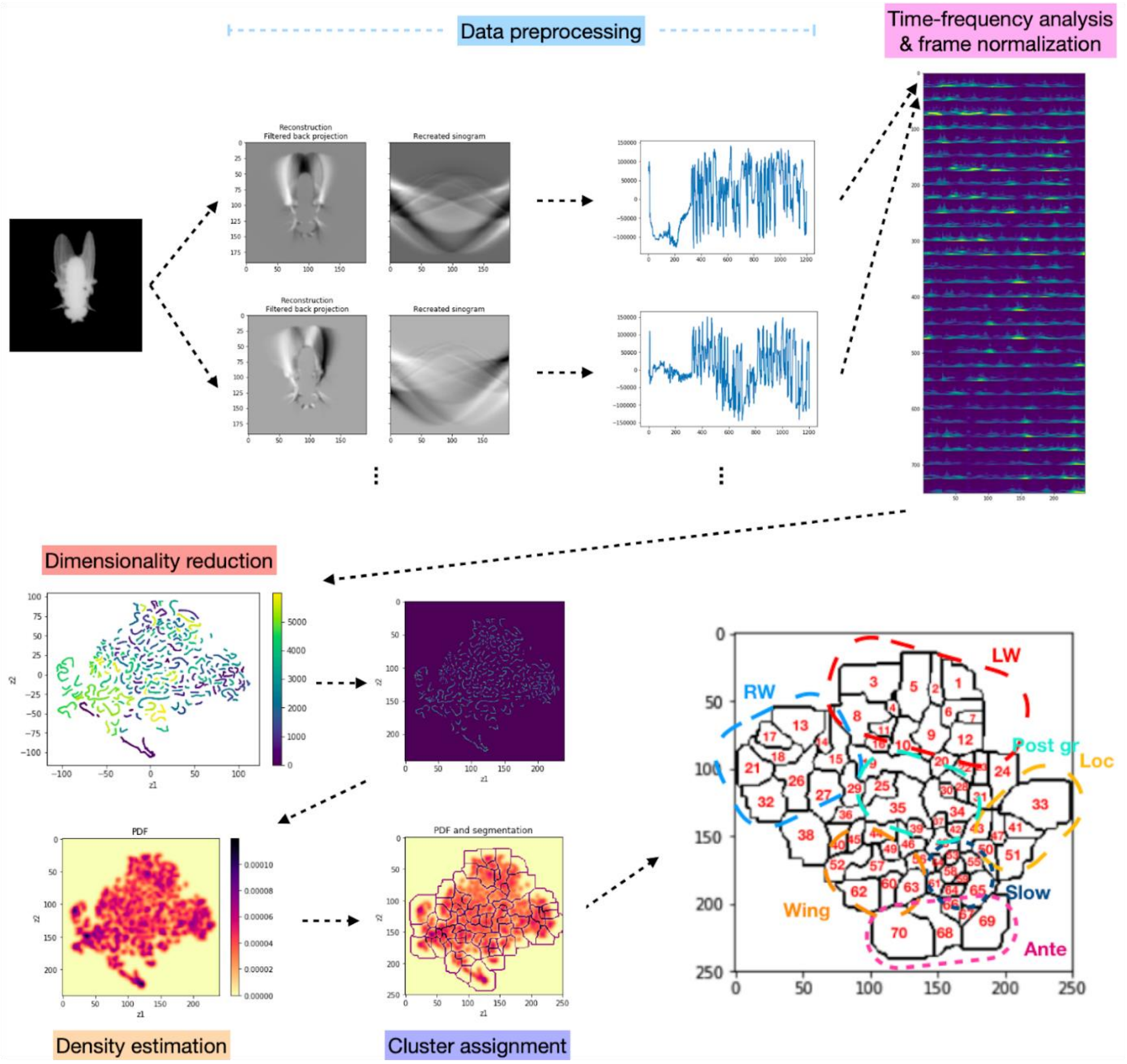


Figure 3.1: Re-mapping stereotyped behaviors from 2D pose *D. Melanogaster* data. Emulating the pipeline in (Berman et al., 2014) in a *Python* setting, with a dataset provided by the author. A Radon transform reshapes each frame's data so that uninformative pixels can be removed. Each spectrogram is reshaped into a 1D-vector, and the dataset is projected onto a 50-dimensional space with PCA. The resulting time series undergo a CWT analysis and are embedded onto a 2-dimensional space with t-SNE. This is followed by a Gaussian-kernel convolution and Watershed segmentation. Likewise, the pipeline successfully cataloged bouts of stereotyped motion patterns in the dataset.

mode (displayed in Figure 3.1). From here, a 2-dimensional representation is achieved using t-SNE. To perform segmentation, the discrete embedded space is smoothed by means of a gaussian convolution kernel, yielding a probability density function (PDF) that estimates the data's underlying distribution. Ultimately, a watershed transform is employed to delineate behavioral regions of similar movement types in the embedded space. The pipeline's manually tuned parameters were chosen in agreement with the ones from the original paper.

At this early stage, the concern for the functionality of the pipeline's building blocks outweighs the concern for the truthfulness of its end results, which are unimportant to this thesis's research questions.

3.3 Adapting the emulated pipeline to 3D pose data

The emulated pipeline described in the previous section requires video data input, which is 2-dimensional, covers merely a dorsal view of the fly, admitting limb occlusions, and promotes computational memory issues and time-consuming algorithms. As stated, *DeepFly3D* achieves rapid, reproducible and comprehensive 3D pose estimation in tethered *D. Melanogaster*, rendering it an ideal substitute for the postural decomposition steps employed in the previous section. In addition, since the pipeline bears a modular architecture, interchanging steps between algorithms is a very feasible task.

3.3.1 Data handling

DeepFly3D is responsible for tracking the *x*-, *y*- and *z*-coordinates of each limb joint, *pretarsus*, abdominal *tergite* stripes and antennal landmarks per time instance, resulting in a 38-dimensional 100Hz time series.

$$\vec{X}(t) \equiv [\vec{x}_1(t), \vec{x}_2(t), \dots, \vec{x}_{38}(t)] \qquad 3.1$$

3.3.2 Error filtering

Before being subjected to analysis, the data traced by *DeepFly3D* was smoothed by means of a *1€* filter (Casiez et al., 2012), a low-pass filter with adaptive cutoff frequency that diminishes jitter at low speeds and latency at high speeds.

3.3.3 Angle-based individual normalization

The filtered data can be converted into angles to achieve translational and individual-scale invariance. Any method that aims to map behavior should be able to do so consistently across separate individuals. This requires a normalization step that attenuates natural morphological individualities, and/or artificial deviations promoted by disparate laboratorial or experimental settings. Variations in fly proportions, and/or relative positions towards the cameras, can theoretically be accounted for by computing angles from sets of connected 3D positions, such as joint, abdominal or antennal angles.

Here, such angles can be calculated by simply taking the dot product of two vectors corresponding to bone segments, antennal segments, or the axis of gravity. To account for a complete set of limb movements, the number of computed angles must match the total number of degrees of freedom for every joint of the fly. The *coxa-femur*, *femur-tibia*, and *tibia-tarsus* hinge joints hold a single degree of freedom while the *body-coxa* is treated as a *ball-and-socket* joint, which holds three degrees of freedom. Disregarding the bending of limbs (which unfortunately occurs, mainly with the flexible *tarsus* segments), the full set of limb joints, along with the *pretarsus* can be treated as belonging to the same 2-dimensional plane. Thus, to approximately describe limb pose, one requires the *coxa-femur*, *femur-tibia*, and *tibia-tarsus* hinge joint angles plus the orientation of each leg's plane. To account for this, a fly reference frame is required. To attenuate differences between experiments, each fly's forward reference vector is determined by averaging the positions of both its hinder and middle *body-coxa* tracked positions and connecting these average coordinates. Accordingly, the plane that holds both the fly's forward reference vector and the gravity axis is known as the fly's *sagittal* plane. Its two perpendicular planes represent the fly's *transversal* and *coronal* planes. Hence, to compute the orientation of a given limb, one needs to take the cross product between the vectors created by their *tibia* and *femur* bone segments, project it onto the three aforementioned planes and take the angles between it and its projected correspondents. In addition to leg related angles, abdominal and antennal angles are extracted from their respective landmarks: the fly's three *tergite* stripes yield three landmarks on each side, which are converted into two abdominal angles; as for the antennae, a third reference landmark is computed by averaging the antennal coordinates across the totality of a recording and employed in tracking the fly's head tilt at every instance - larger angles indicate smaller deviations from the head's rest position.

Presently, more sophisticated methods that rely on refined mathematical tools such as rotation matrices, Euler angles and quaternions are available, achieving more consistent and truthful results than the abovementioned approach. Such a method (based on *Homogeneous Transformation Matrices and Quaternions*) is included as an alternative to this stage of the data preprocessing, but is nonetheless dismissed throughout the remaining sections of this thesis. The development of a state-of-the-art method for angle normalization would be prohibitively time consuming when effective options are already in use.

The 6 angles per limb (top of Figure 3.2), plus 2 abdominal angles and 1 antennal angle add up to 33 angle time series. Nonetheless, as assumptions are made, biases are possibly introduced, and some information might be misinterpreted or lost. Consequently, this sort of normalization is only advised when combining datasets from multiple individuals with morphological disparities and can otherwise be overlooked.

3.3.4 Time-frequency analysis with a continuous wavelet transform

The next step concerns the dynamical aspect of behavior, since a mere postural description of behavior would remain incomplete. Here, time-frequency analysis using a *Continuous Wavelet Transform* (CWT) is preferred over motif-finding techniques, due to the multi-resolution time-frequency trade-off and absence of temporal alignment surplus (middle panel of Figure 3.2). The Morlet wavelet, defined as the product of a complex sine wave and a Gaussian window, was selected for CWT, due to its suitability for isolating chirps of periodic motion (Equation 3.5).

$$\psi(t) = \pi^{-1/4} e^{i\omega_0 t} e^{-1/2t^2} \tag{3.2}$$

Initial efforts were carried out with *Python*’s *PyWavelets* open-source package, though limitations regarding lower frequency scales and flexibility of tuning parameters prompted the recreation of the wavelet transform carried out by Berman et al., (*2014*), within a *Python* environment. This transform operates according to Equation 2.1 for CWT, while dividing the signal by a scalar factor ($C(s)$ from Equation 3.5) to correct for disproportionally large responses by lower-frequency signals.

$$C(s) = \frac{\pi^{-1/4}}{\sqrt{2s}} e^{1/4\left(\omega_0 - \sqrt{\omega_0{}^2+2}\right)^2} \tag{3.3}$$

The recreated wavelet function is very adaptable in terms of number of frequency channels, or minimum and maximum targeted frequencies: their default values are set at 25 frequency channels spaced logarithmically between 3Hz and 50Hz, the Nyquist frequency of this system (since the cameras’ sample rate is 100Hz).

After performing the CWT and preserving only the magnitudes of the transform, a 825-dimensional (33 x 25 = 825) time series, $S(k,f;t) \in \mathbb{R}^2$, emerges from the initial 33-dimensional postural time series, where the amplitudes reflect the power at each frequency, for each postural mode, within the window of a corresponding scale (which translates from the frequency channel, and vice-versa) (Equation 3.4, where $s(f)$ is given by Equation 3.5). Since dynamical information was added to the fly’s postural space, the newly generated space can be referred to as a posture-dynamics space.

$$S(k,f;t) = \frac{|W(s(f),t;\alpha(\tau),\psi(\tau))|}{C(s(f))} \tag{3.4}$$

$$s(f) = {\omega_0 + \sqrt{\omega_0{}^2+2}}\Big/{4\pi f} \tag{3.5}$$

3.3.5 Low energy frame rejection

This step is required to account for the subsequent step of frame normalization, where low-level noise energy is amplified in resting frames. All the rejected low energy frames are set aside to a single cluster, preventing later erroneous assignments or the creation of multiple, redundant rest clusters. The dismissal of data frames relies on the variance distribution of each recording’s power spectrum, $S(k,f;t)$, which often presents itself as a bimodal distribution (Figure 4.1(b)). Thus, each frame can be labeled either as an *active* frame or *resting* frame, depending on whether the variance of its frequency channel amplitudes surpasses a threshold value determined by Otsu’s method (Otsu & N., *1996*).

Here, Otsu’s threshold is computed by means of a *scikit-learn* function and the *resting* frames are stored for later recreation of videos that corroborate the fly’s absence of activity, while the *active* frames proceed to the ensuing stages of the pipeline.

3.3.6 Frame normalization

To account for the finite nature of the wavelet’s analysis window, each frame within the posture-dynamics’ expression matrix is normalized by the sum of wavelet magnitudes from every mode-frequency channel (i.e. along the columns of $S(k,f;t)$) (Equation 3.6). The resulting 825-dimensional

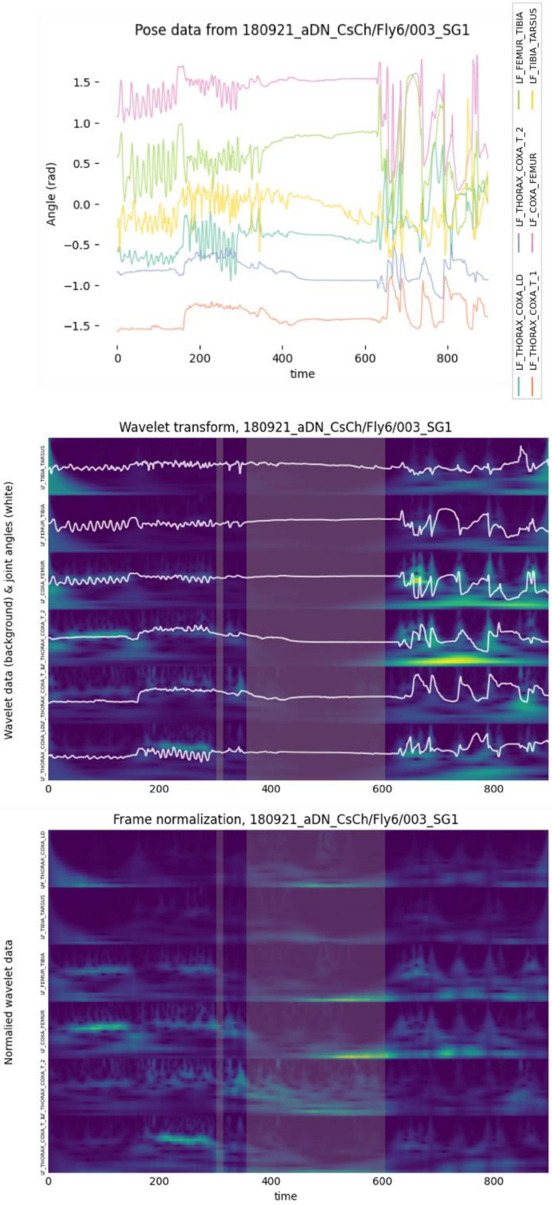


Figure 3.2: Data preprocessing. Following feature angle normalization (top panel), pose data undergoes a CWT analysis with 25 Morlet wavelets, targeting signal frequencies between 3 and 25Hz. This yields an 825-dimensional posture-dynamics expression matrix, $S(k, f; t)$ (middle panel). *Active* or *resting* frames are labeled according to the variance of the columns of $S(k, f; t)$ (*resting* frames are shaded in grey). Lastly, the columns of $S(k, f; t)$ are normalized as to resemble positive semi-definite probability distributions ($\hat{S}(k, f; t)$ in the bottom panel).

space, $\hat{S}(k, f; t)$, consists of an ensemble of vectors that can be thought of as probability distributions over all mode-frequency channels, at a given point in time (frame) (bottom of Figure 3.2).

$$\hat{S}(k, f; t) = \frac{S(k, f; t)}{\sum_{k', f'} S(k', f'; t)} \qquad 3.6$$

3.3.7 Dimensionality reduction (with t-SNE)

This step is concerned with projecting high-dimensional data points, possibly lying within a lower-dimensional manifold, onto a 2-dimensional space, deeming it observable and inviting to a segmentation step. As stated in Section 2.4.1.1 , t-SNE is an ideal embedding candidate since it preserves local structure, which is key when looking for shorter-scaled stereotyped trajectories – which translate to subtle, but consistent moves performed by the animal.

The t-SNE algorithm is implemented with *Python*'s *scikit-learn* data analysis package, in a straightforward fashion. Testing several combinations of the algorithm's adjustable parameters (such as *perplexity*, *early exaggeration*, *number of iterations*, *learning rate*…) reveals that the topmost influential parameters are *perplexity*, the *distance metric* and, to some extent, the *number of iterations* (which is only meaningful if it is too low; otherwise, it can be exaggerated at just the cost of time efficiency). Hence, the remaining parameters are left to their standard values, while the crucial ones are determined through the casting of semi-empirical laws. The *perplexity* value is set according to an empirical power law - *perplexity* ~ $N^{1/2}$, where *N* is the number of data points to embed - which is backed by the intuition that *perplexity* reflects how many points are perceived by each data point. For this dataset, the *perplexity* value revolves around the [125,150] interval. Regarding the *distance metric*, Berman et al., (2014) take advantage of the vector normalization step to treat each mode-frequency feature vector from $\hat{S}(k, f; t)$ as a probability distribution, at a given point in time, *t*. Hence, the Kullback-Leibler (KL) divergence can be considered as the algorithm's *distance metric* (although, technically, KL is a divergence - not a metric) when calculating distance between feature vectors (Equation 3.7). For reference, the KL divergence is an asymmetrical measurement of how different two probability distributions are from one another.

$$d(t_i, t_j) = D_{KL}(t_i \| t_j) = \sum_{f,k} \hat{S}(k, f; t_i) \log_2 \left(\frac{\hat{S}(k, f; t_i)}{\hat{S}(k, f; t_j)} \right) \qquad 3.7$$

3.3.8 Density estimation (with a Gaussian convolution kernel)

Occasionally, an intermediate density estimation step interposes dimensionality reduction and clustering assignment of data elements. The intent is to convert the discrete embedded space onto a continuous *probability density function* (PDF) and make way for clustering algorithms otherwise incompatible with discrete space inputs. Prevalent techniques for density estimation include Gaussian convolution kernel convolutions and GMM.

Following the normalization of t-SNE coordinates, a mesh (500x500 *px*) is set up to collect the embedded data points so that density estimation and segmentation steps can be performed upon them (top-left in Figure 3.3). At this stage, density estimation is performed by means of a Gaussian kernel convolution (*σ=4px*), yielding an estimate of the PDF of the embedded plane, $P(z_1,z_2)$ (top-right in Figure 3.3). Local peaks in $P(z_1,z_2)$ usually portray stereotyped behaviors - the exception being isolated (i.e. with very few near neighbors) data points, whose prevalence depends on the number of total embedded data points, mesh dimensions and *σ*. Since they are so decisive to the outcome of this pipeline, the mesh size and Gaussian kernel size are treated as key hand tuned parameters. Unfortunately, this pipeline still lacks success benchmarks for these parameter's values, which must be assigned according to the user's intuition and experience. A combination of higher mesh sizes and low *σ* leads to coarser clusters, while the opposite leads to more refined clusters - both error prone when these values surpass a reasonable limit.

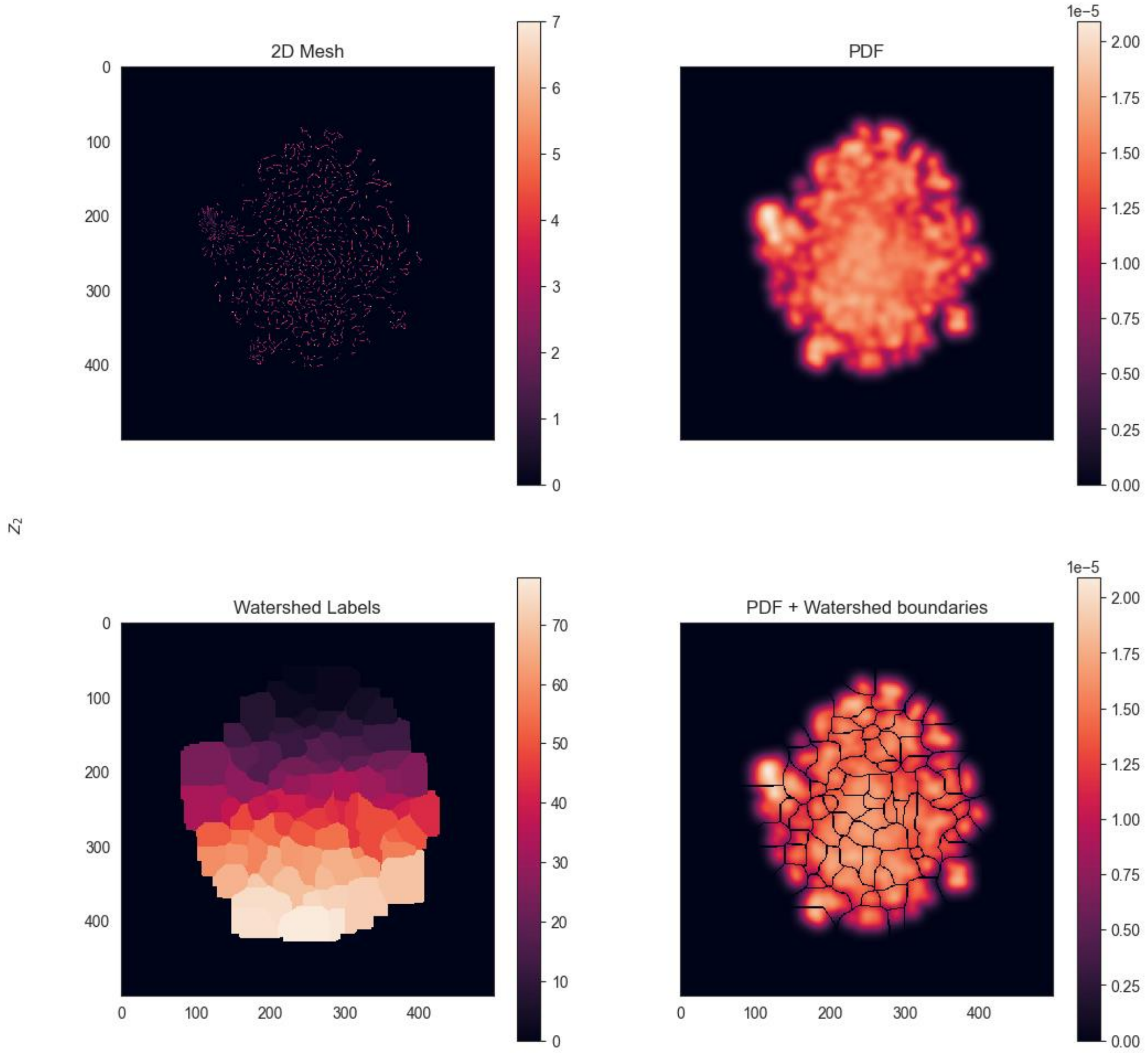


Figure 3.3: Segmentation of the embedded posture-dynamics space. An empty matrix (500x500) is created to accommodate the embedded data points (top-left panel). Then a Gaussian-kernel convolution is employed (top-left panel) and followed by Watershed segmentation (bottom panels). Each area in the bottom right panel holds a single PDF maximum and represents a stereotyped behavior.

3.3.9 Cluster assignment (with a Watershed transform)

Any two given local maxima of $P(z_1,z_2)$ must be separated by a valley and, inversely, climbing the gradient of $P(z_1,z_2)$ from any given point in a valley will lead to a local maxima. Watershed segmentation (Meyer, 1994) makes use of this property to partition $P(z_1,z_2)$ into areas where climbing the gradient of $P(z_1,z_2)$, starting at any (z_1,z_2) within them, leads to the same local maxima which, in turn, represents a given stereotyped behavior (bottom of Figure 3.3).

The watershed algorithm is deterministic and inflexible regarding its outcome under different combinations of its specifications, the sole exception being the *minimum distance* parameter, whose influence over the pipeline's final outcome is only noticeable with low Gaussian kernel size (σ) values

(Figure 4.3(a)). Hence, watershed segmentation is implemented under standard parameter values, with *Python*’s *scikit-image* package.

3.4 Modular alterations of the default pipeline

The previous pipeline was mainly based on that of Berman et al., (*2014*). Nonetheless, since the conception of this groundbreaking work, alternative algorithms fitting within the modular architecture of its pipeline have been developed. Furthermore, since datasets differ between video and key-landmark time series, exploring several combinations of embedding and/or clustering algorithms might uncover an optimal pipeline for unsupervised classification of behavior from 3D pose data. Todd et al., (*2017*) compared unsupervised methods for mapping behavioral spaces from *Drosophila*, according to specific, desirable properties of their clustering outputs, such as *entropy*, *uncompactness* or *mean dwell time*. t-SNE competed with PCA at the dimensionality reduction stage, while for the steps of density estimation and clustering, GMM-posterior probability assignment was compared to 2D-Gaussian blur-watershed. Performing PCA onto 20 dimensions, followed by GMM and sparse-watershed cluster consolidation proved to be the best mapping method, by their quality standards.

Likewise, this section exploits clustering and embedding variations by introducing modular variations among clustering and embedding algorithms. t-SNE and PCA remain as prime embedding candidates, while GMM and 2D Gaussian blur perform density estimation, and clustering is left to posterior probability assignment and HDBSCAN algorithms. Due to its nature, HDBSCAN can perform clustering in high dimensional spaces, which will be further explored. The schematics of all combinational pipelines is given in Figure 3.4.

3.4.1 Dimensionality reduction (with PCA)

Unlike t-SNE, PCA favors the preservation of global space structure, by projecting the data onto an orthogonal basis composed by the covariance matrix’s eigenvectors. While this renders PCA impractical for isolating local trajectories that reflect stereotyped behavioral bouts, it may prove useful for direct visual comparison with t-SNE, when embedding the data onto 2 dimensions, and as an intermediate step for soothing the computational demands of clustering in high-dimensional spaces.

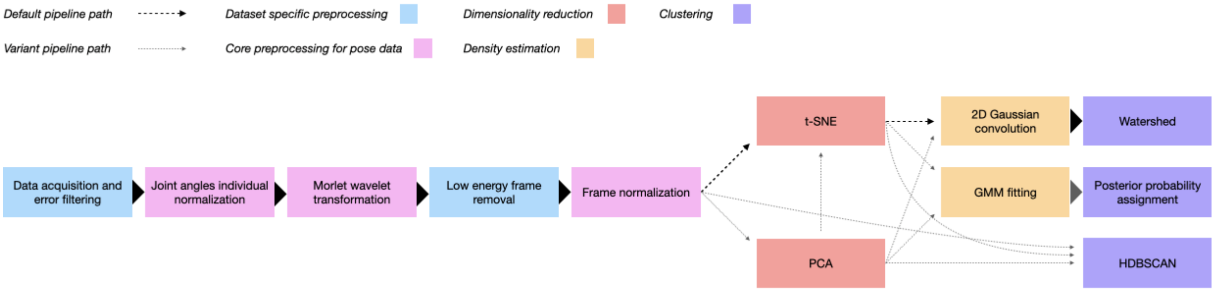

Default pipeline path
Variant pipeline path
Dataset specific preprocessing
Core preprocessing for pose data
Dimensionality reduction
Density estimation
Clustering
Data acquisition and error filtering
Joint angles individual normalization
Morlet wavelet transformation
Low energy frame removal
Frame normalization
t-SNE
PCA
2D Gaussian convolution
GMM fitting
Watershed
Posterior probability assignment
HDBSCAN

Figure 3.4: Flow chart diagram of every combination of modules for mapping behavior. Darker arrows indicate the route of the default pipeline (emulated from in (Berman et al., *2014*)), from which the remaining candidates arise by interchanging modules for dimensionality reduction, density estimation and clustering (grey arrows).

PCA is computed with *scikit-learn* and the sole manageable hand tuned parameter refers to the number of components to keep. When PCA serves an intermediate step for high-dimensional clustering, the aim is to keep this number as low as possible, while retaining no less than 85% of the explained variance of the dataset - which translates roughly to 30 to 50 principal components for the current dataset (depending mainly on the number of points that go through the embedding step). When PCA is employed for mapping purposes, only 2 principal components are preserved, meaning the number of components to keep is no longer regarded as a key hand tuned parameter.

3.4.2 Density estimation (with GMM)

This technique fits the data with a combination of mixture components, multivariate gaussian distributions, and iterates their parameters (weights, means, variances) until convergence is attained. Like the 2D Gaussian kernel technique, GMMs demand a single hand-tuned parameter - the number of mixture components, noted *k*, which translates to the number of identified clusters (none is left empty). Thus, the *a priori* assessment of the number of clusters poses a critical challenge. A common approach involves the performing of an exploratory analysis with some unsupervised approach (e.g. t-SNE-watershed) to determine the number of clusters in the data, which translates to the value of *k*. Alternatively, one can rely on two statistical criteria for model selection: the *Akaike Information Criterion*, AIC, and the *Bayesian Information Criterion*, BIC. The former is, defined by Equation 3.8 (where $\hat{L}$ is the maximized value of the model's likelihood and *d* is the number of parameters estimated by the model) and estimates the amount of relative information lost by a given model, contemplating its risks of over- or underfitting; BIC is formally defined by Equation 3.9 (where *N* is the number of data points in the experiment) and is closely related to AIC. For both criteria, lower values are indicative of a better model. Thus, to assess the appropriate value of *k*, one can systematically adjust its value and find the knee of the AIC/BIC-k curve (Figure 4.8).

The GMM leaves the data points undisturbed, while yielding a matrix of posterior probabilities that ties each data point to each mixture component.

$$AIC = 2d - 2\ln(\hat{L}) \tag{3.8}$$

$$BIC = d\ln(N) - 2\ln(\hat{L}) \tag{3.9}$$

3.4.3 Cluster assignment (with posterior probability assignment)

This simple clustering assignment technique is only enforced in continuity with GMM, making use of its outputted posterior probabilities matrix that assigns each data point with a set of probabilities of drawing said point from each mixture component. Computationally, posterior probability assignment plainly means to label each data point according to the mixture component that presents the larger posterior probability. As remarked by Berman et al., (*2014*), behavioral spaces may comprise clusters of reflecting stereotyped behaviors bounded by less dense regions associated with non-stereotyped actions. Hence, although issues may arise with data points situated between mixture components, stereotyped trajectories are left, for the most part, undisturbed. Posterior probability clustering yields a 1-dimensional vector where each entry denotes the cluster number attributed to the corresponding frame.

3.4.4 Cluster assignment (with HDBSCAN)

HDBSCAN (Campello et al., 2013) is a non-parametric method that extends from the aforementioned DBSCAN, which offers a favorable matchup against clusters of varying shapes and, nonetheless, struggles with density-heterogeneous clusters. While making very few assumptions about the clusters, HDBSCAN attempts to separate denser regions from their surrounding space and establish a cluster hierarchy. Its properties render it ideal for dealing with noisy data containing clusters of arbitrary shapes, different sizes and densities.

The algorithm can be summarized in five steps. First, it transforms the data space by rearranging its points according to a *mutual reachability distance* metric - dense points conserve their relative distance while sparser (noisy) points are further isolated. Then, HDBSCAN proceeds to build a *minimum spanning tree*, a simplified graph where links between data points are weighted under the *minimum reachability distance*. Subsequently, the links are sorted by their distance to create a hierarchy of connected components, resulting in a dendrogram. To achieve a set of flat clusters of varying densities, HDBSCAN further condenses the dendrogram into a smaller tree with more data points attached to each node. Finally, the algorithm works up the tree, comparing the stabilities of clusters and their direct descendants at each tree branch - higher stability clusters are selected, and their descendants unselected until the root node is reached.

Once more, the employed HDBSCAN function is imported from *scikit-learn* and implemented under a tolerant *minimum cluster size* of 20 frames, motivated by the threshold for outlier sequences upon the video reconstruction step (10 frames) - if a behavioral bout ought to comprise at least 10 consecutive frames of data, thence, a cluster, which should cover at least two embedded trajectories, should hold at least 20 data points. Increasing the *minimum cluster size* parameter reduces the number of labels, while merging some of the clusters together. Another important parameter to balance *minimum cluster size* is the *minimum samples* parameter, which reflects how conservative the algorithm is: the higher its value, the likeliest it is for data points to be considered as noise, resulting in fewer labeled clusters. For further note, HDBSCAN admits additional parameters that are, nevertheless, left undisturbed.

Clustering with HDBSCAN was employed directly over the posture-dynamics expression matrix ($\hat{S}(k,f;t)$) and its embedded version (2-dimensional space). Although the low dimensional application of HDBSCAN is straightforward, in a high dimensional space, HDBSCAN should be preceded by any sort of dimensionality reduction, due to the infamous *curse of dimensionality*. Here, the selected candidate was PCA onto 125 dimensions, mainly due to its promptness.

3.5 Creating video sequences from clustering results

After their respective segmentation steps, both the default pipeline and its derivatives converge onto a segment where sequential frames sharing the same cluster label are assembled into video sequences, serving purposes of clustering validation. Human observation-based validations would technically revert the methodology back to *supervised* classification, since the researcher's interpretations are subjected to bias as well. Nonetheless, the main purpose of this step is to provide the user with interpretable results, allowing for a primary assessment of cluster quality, which should be subsequently reinforced by means of unbiased performance metrics (Todd et al., 2017).

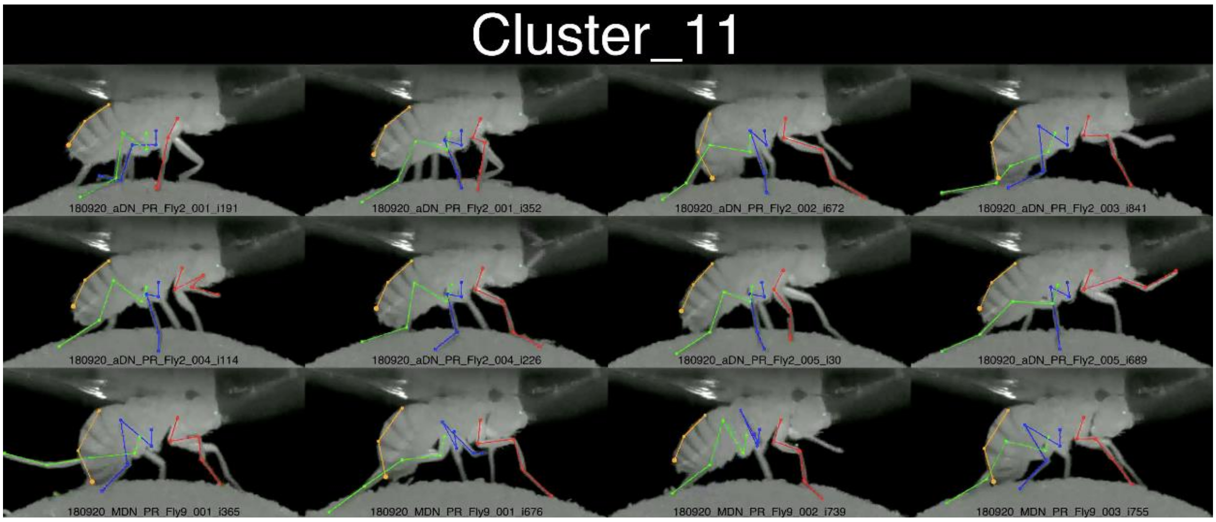

Cluster_11
180920_aDN_PR_Fly2_001_i191
180920_aDN_PR_Fly2_001_i352
180920_aDN_PR_Fly2_002_i672
180920_aDN_PR_Fly2_003_i841
180920_aDN_PR_Fly2_004_i114
180920_aDN_PR_Fly2_004_i226
180920_aDN_PR_Fly2_005_i30
180920_aDN_PR_Fly2_005_i689
180920_MDN_PR_Fly9_001_i365
180920_MDN_PR_Fly9_001_i676
180920_MDN_PR_Fly9_002_i739
180920_MDN_PR_Fly9_003_i755

Figure 3.5: Cluster videos output. Each panel holds a video sequence comprised of a succession of frames from the same cluster, with the prerequisite that the sequence is at least 10 frames long (otherwise it is labeled as an outlier sequence). The panels also include bone segment markers, the fly's label and the initial frame number, as to ease interpretation.

Accordingly, a dictionary is created to retain the information of the low dimensional space (such as data point coordinates and frame activity values) and sort data by cluster, identity (genotype and fly ID included) and behavioral, or outlier, sequence number (chronological order of each cluster's stereotyped actions). Each data point is linked with its source frame from one of the 7 cameras used by *DeepFly3D* and cluster videos are created from non-outlier frame sequences that comprise each cluster in the embedded space. To achieve this, a complex function employs a cluster-specific *ffmpeg* command, then called by means of a *Python* subprocess to yield a mosaic of video sequences marked with the cluster label number (Figure 3.5). In addition, each casted fly is identified by its name tag and its bone segments are conveniently marked with different colored lines for each limb to facilitate ocular tracking.

3.6 Benchmarks for clustering success

To help assess each pipeline's strengths and shortcomings, and overall clustering success, visual inspection of the videos is coupled with unbiased metrics that evaluate embedding and clustering quality. These metrics are said to be *unbiased* in the sense that they are reproducible across any behavior clustering algorithms beyond the scope of this thesis. They tend to specific ambitions of this behavior clustering endeavor: 1) clusters must comprise a discernible pattern of behavior among its constituent trajectories; 2) clustering shouldn't reflect the identity of the individual - a step above from this would be the meaningful clustering of behaviors from individuals of different species; 3) clusters should be balanced and transitions shouldn't be nor too predictable, nor too random; 4) the clustering pipeline ought to be unbiased, comprising as few hand tuned parameters as possible; 5) runtime shouldn't be too lengthy - which would be impractical.

With each pipeline run, three metrics are computed from the information carried by the aforementioned clustering results dictionary. *Fly homogeneity* is an embedding metric that evaluates how evenly the data from different flies is embedded (Equation 3.10). To quantify this, a distance matrix, noted C_{ij}, is computed from centroid coordinates of each fly's embedded data points and then compared with a reference matrix, C_{ij}*, via the KL divergence (Equation 3.7). Here, the reference matrix is created by randomly shuffling the columns of the original distance matrix, as to dissipate the influence of identity or genotype. Although not an ideal reference (it is hard to estimate what an ideal centroid distances distribution looks like), it still offers interpretable results. *Entropy* (Equation 3.11) and *mean dwell time* (Equation 3.12) were devised in a similar fashion to the homonymic metrics by Todd et al., (2017), with the former being a measure of how evenly the data points are distributed across the labeled clusters and the later concerning the mean number of frames the fly spends occupying a cluster before transitioning to another one. Lower *entropy* values point to unbalanced frame distributions, where a few clusters are favored and others avoided, which is considered as an undesirable property for suchlike clustering endeavors. *Mean dwell time* is a descriptive metric and comes with no universal standard of success. However, visual inspection of the dataset's videos, suggests natural behavioral bout durations taking between 10 and 30 frames (0.1-0.3 seconds, at f_s=100Hz) which will set the barometer for this metric. The outlier threshold is set at 10 frames, thus any clustering that yields a *mean dwell time* that lies under this lower bound is considered highly undesirable.

$$Fly\ homogeneity = D_{KL}\left(\boldsymbol{C_{ij}} \middle\| \boldsymbol{C^*_{ij}}\right);\ \boldsymbol{C_{ij}} = \left\|c_i - c_j\right\|^2 \qquad 3.10$$

$$Entropy = -\sum_{k=1}^{N_{clusters}} p_k \log_2(p_k) \qquad 3.11$$

$$Mean\ dwell\ time = \sum_{s=1}^{N_{sequences}} \frac{len(s)}{N_{sequences}} \qquad 3.12$$

Besides these metrics, two more criteria help assess whether the pipeline conforms to this endeavor's objectives: the number of hand tuned parameters (parameters that have a decisive influence over the outcome of the algorithm), which ought to be as low as possible if the pipeline is to be considered unbiased, and the pipeline's run time.

Clustering pipeline evaluations were conducted by running each pipeline a total of eight times and comparing the quality/accuracy of their cluster videos and metric ranks (Table 4.1). A more thorough analysis of cluster quality would require additional meaningful metrics, which, although in reach, were not the primary focus of this thesis and would demand further time.

3.7 *Drosophila* experiments

To test the influence of features unrelated to behavior, such as changes in camera perspectives or inter-animal morphology, unsupervised behavioral classification was performed on video data from optogenetic stimulation techniques that consistently drove to specific, documented actions. As described by Günel et al., (2019), this dataset consists of optically activated *MDN>CsChrimson* flies that favor backwards walking (or *moonwalking*), *aDN>CsChrimson* flies that tend to perform antennal grooming, as well as control *MDN-GAL4/+* and *aDN-GAL4/+* flies, that lack the *UAS-CsChrimson* transgene. Nevertheless, only the fraction of this dataset's elements that include pose data and video data concurrently is employed while evaluating the default and modified behavior classification pipelines. This subset consists of 19 trials of *aDN>CsChrimson* flies and 5 trials of each control *MDN-GAL4/+* and *aDN-GAL4/+* flies.

3.8 Data and code

The data and code used in this research were uploaded to a GitHub repository and are available for download at *https://github.com/JoaoCampagnolo/Behav_clustering_thesis*.

3.9 Materials (hardware and software)

The analysis was carried out in a *Jupyter Notebook* environment with custom-built *Python* (version 3.7.2) functions, on a MacBook Air with a 1,6 GHz Intel Core i5 CPU and 4 GB of DDR3 RAM. For a dataset of 25,000 data points from behaving *D. Melanogaster*, run times vary between 3 and 22 minutes, depending on the combination of modules, if the recreation of video sequences is discarded.

Chapter 4

Results and discussion

4.1 Effectiveness of angle normalization

All the aforementioned pipeline modular combinations share a common path regarding data their data preprocessing, joint angle normalization, wavelet transformation and frame normalization steps (later steps already shown in Figure 3.2). The 3D datasets provided by *DeepFly3D* make way for unprecedented feature angle normalization across multiple *D. Melanogaster* individuals in unsupervised behavioral mapping techniques. To assess the significance of this step, *t-SNE*$_2$*-Watershed* clustering was performed a total of 20 times for a dataset consisting of 13 trials from 7 different control *aDN-GAL4/+* flies, with and without joint angle normalization (while keeping key hand tuned parameters at constant values) and the resulting spatial distributions of embedded data points from individual flies were compared in terms of proximity and shape. Unfortunately, video data was not available for this dataset at the time this analysis was performed, rendering result validation from cluster video observations problematic, though not impossible, since the videos are available along with frame labels at the data and code repository (Section 3.8).

Proximity was evaluated by taking the mean coordinates of each fly's embedded data points (further referenced as *fly cores*) and calculating pairwise Euclidean distances across them. The spatial distribution of each fly's embedded data points was surveyed by an *uncompactness* metric, analogous to that used by Todd et al., (*2017*), by computing each fly epicenter, calculating the mean distance of every fly's embedded data points to the *fly core* and averaging this value across all flies.

Conceptually, subject normalization suggests that pose-wise similar data points from different flies are embedded closer together, despite the morphological dissimilarities of the animals from which they come from. It was therefore expected at first glance that the data points embedded under feature angle normalization would present lower pairwise distances between *fly cores* and higher *uncompactness* values among each fly, assuming the flies perform similar behaviors. Nonetheless, this is contradicted by the information in Figure 4.1(c), as *uncompactness* values decrease and pairwise distances between *fly cores* increase when joint normalization is performed. *Uncompactness* loss is depicted clearly in Figure 4.1(a), for *aDN_PR_Fly6*, whose active frames amount to two clusters when t-SNE is performed without joint angle normalization, but only one, when t-SNE is performed after its use. The observation of *aDN_PR_Fly6*'s video recording shows bouts of forward walking and odd brushes of the ceiling with the right foreleg (see dataset directory in Section 3.8). Activity wise, this recording is largely dominated by *rest* frame labels - the first ~100 frames make up for all the *active* frame labels and, when normalization is employed, only an uninterrupted subsequence of ~10 frames remain. This comes as a consequence of a stricter threshold for deeming a given frame as active at the preprocessing stage - active frames dropped from 10890 to 8570 (~21%) with the introduction of feature angle normalization, which was followed by a drop in the number of clusters (Figure 4.1(b)). Accordingly, less *active* frames imply fewer possible behavioral bouts are represented after dimensionality reduction, i.e. fewer regions of the t-SNE map are occupied by each fly, which leads to lower *uncompactness* values. The same occurs for epicenter pairwise distances: with the suppression of some behavioral bouts, each fly places less data points throughout the embedding space, meaning more compact data point distributions for each fly; consequently, *fly cores* will be further displaced from the center of the embedding space, and their pairwise distances will increase.

The decrease of embedded data point numbers does not necessarily mean suppression of shorter behavioral bouts - in most instances it translates to the premature elimination of data points that otherwise would end up labeled as outliers after the segmentation stage. In particular, for *aDN_PR_Fly6*, both approaches yielded a single cluster as final output, since the extra cluster that stemmed from the embedding of non-normalized pose data was composed of short sequences of frames, below the 10 frame threshold for outlier frame sequences. Among them, the clusters differed solely on the number of frames, but expressed the same behavioral bout, nonetheless. Plus, the decrease in data points yielded faster t-SNE embeddings - ~2.5 minutes saved for this relatively small dataset of *aDN>PR flies* - given that the t-SNE function employs a Barnes-Hut approximation that runs in *O(N.Log(N))* time, where *N* represents, again, the number of data points to embed.

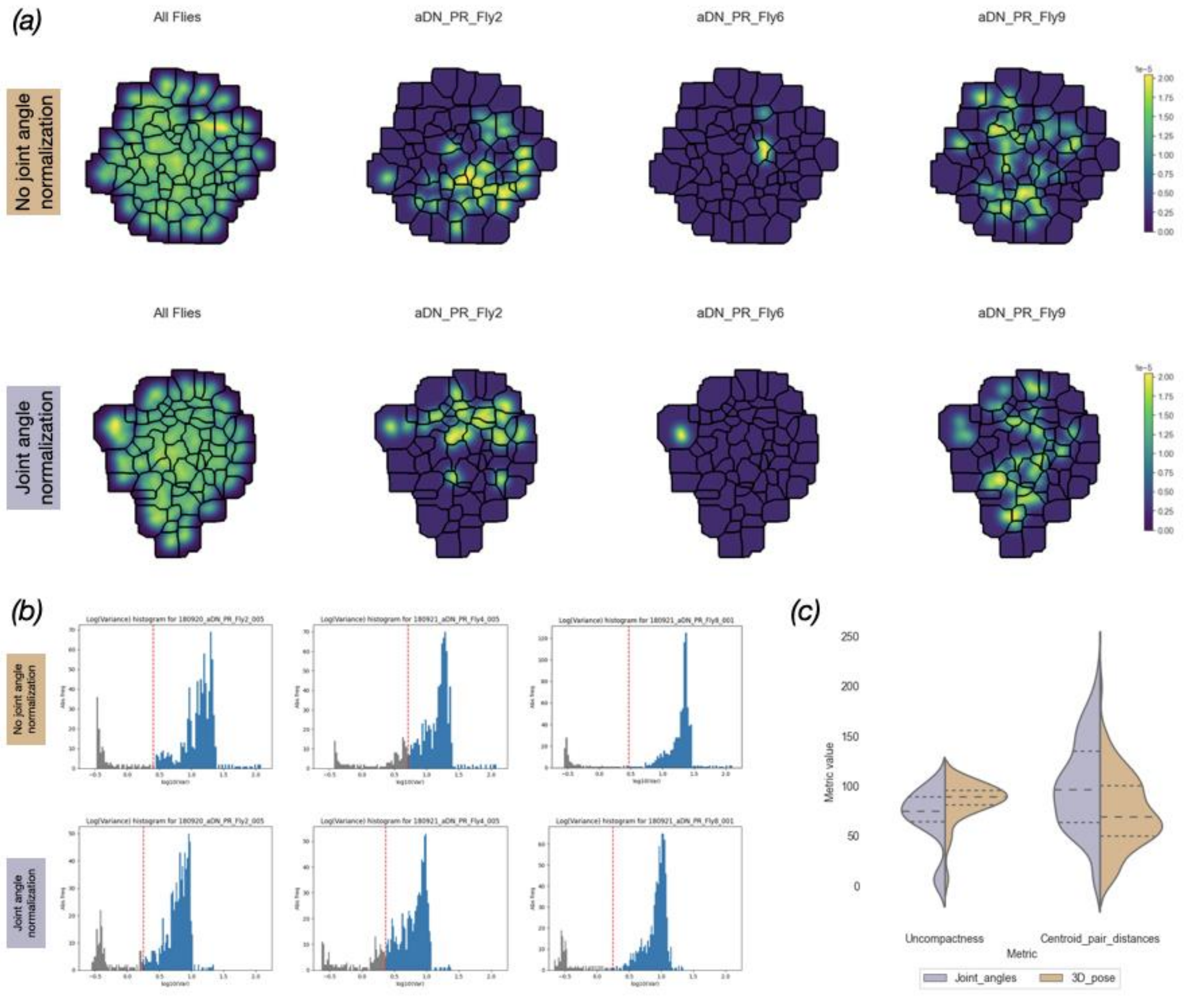


Figure 4.1: Influence from feature angle normalization. **(a)** Comparing embedding methods with and without the inclusion of feature angle normalization. Flies #2 and #9 occupy similar regions of the embedded space, while fly #6 presents an additional cluster when feature angle normalization is ignored. **(b)** Comparing frame activity distributions: in both cases, the distributions are often bimodal (thus facilitating the distinction between *active* and *resting* frames). The relative proportion of resting frames increases with the inclusion of feature angle normalization. **(c)** Comparing the distributions of fly-specific data points in the embedded space: shape is evaluated by *uncompactness* and fly proximity by core pairwise distances. The counterintuitive results (increased fly-core distances and decreased *uncompactness* with angle normalization) are owed to a ~20% decrease in the number of *active* frames, which was retrospectively corrected (supplementary results in Section 3.8's repository).

From the user's perspective, joint angle normalization seems to pose a trade-off between coarser, raw depictions of captured behaviors and faster, more pragmatic portrayals of behavior. Furthermore, if the user can sacrifice temporal efficiency in its entirety, the option to ignore rest and active frame labels is available, and dimensionality reduction can proceed with the full set of data points. A more complete assessment of this step's relevance over the equitable embedding of frames from different animals would require at least disabling the activity filter (Section 3.3.5), as to guarantee the analysis of unvarying frames. A dataset containing multiple instances of the same behavior being performed by distinct flies would further improve the effectiveness of this research. Unfortunately, at this instance, no such dataset is available, nor is there time to repeat the experiment with a disabled activity filter.

4.2 Clustering with the default modular configuration (*t-SNE$_2$-Watershed*)

The above-named "default" pipeline (alternatively noted as *t-SNE$_2$-Watershed*) was tested with the dataset that features pose and video data from 28 trials of *aDN>CsChrimson*, *MDN-GAL4/+* and *aDN-GAL4/+* flies (Section 3.7). In (Berman et al., *2014*), nearby embedded data points show similar activity values, despite the presence of a normalization step for each frame's total spectral power. Nonetheless, two exceptions to this principle stand out in this research: foremost, a mixed activity cluster, holding the topmost active data points and characterized by a fairly low *dwell time* (when compared to other clusters), persists at a given extremity of the low dimensional space (Figure 4.2). Inspection of its cluster videos reveals a variety of behaviors, from resting to grooming and ball pushing. Further examination of the results dictionary shows the constituent frames arrive exclusively from either the first or last 20 frames of every represented fly recording. These cluster elements received an unexpected boost in frame power spectrum variance due to their proximity to the recording's bounds, as portrayed in Figure 3.2. Remarkably, the t-SNE function is still able to group all these boundary frames together, preventing them from polluting other legitimate clusters.

The second predicament arises from the t-SNE function itself and the information carried by low frequency-targeted wavelet transforms. As lower frequencies are considered in wavelet analysis, trajectories in the embedded spaces gradually assume a form of elongated strips of points (further

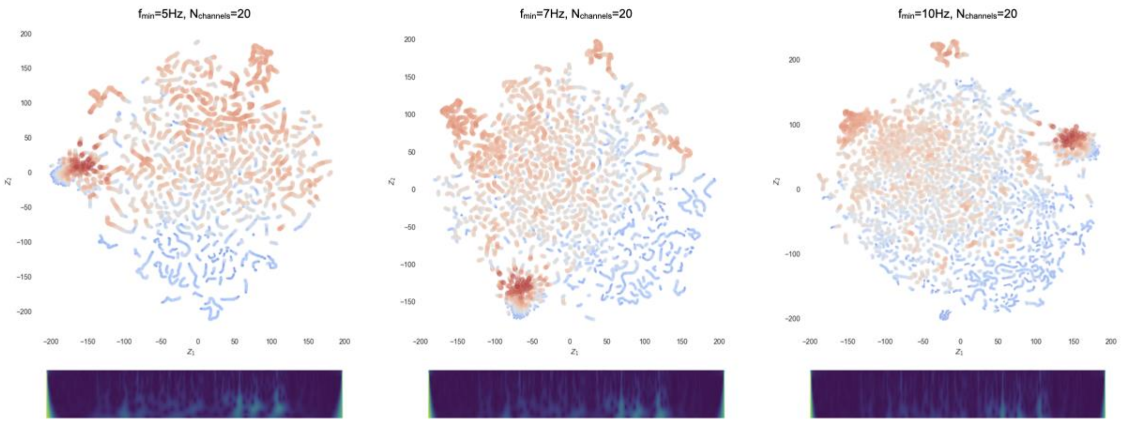


Figure 4.2: Embeddings under different CWT analysis minimum frequencies. When lower frequencies are included (bottom panels), trajectories in the embedded space become longer, comprising additional discernible patterns of behavior. Furthermore, all the embeddings feature a mixed activity cluster, composed by the frames that come from the recording's edges.

mentioned as *vermiform trajectories*) (Figure 4.2), rather than the disconnected conglomerates observed by Berman et al., (2014). This phenomenon persists even when varying other t-SNE parameters such as *perplexity*, *number of iterations*, *learning rate*, and *pairwise distance metric*. In the posture-dynamics space, power spectrum vectors from neighboring frames show greater similarity when slow varying low frequency channel signals are included. This increased similarity then reflects in a shorter distance between the embedded neighboring frame vectors, after t-SNE. In short, when low frequency channels are included in time frequency analysis, the posture-dynamics space, $\hat{S}(k, f; t)$, becomes too smooth for the fastidious t-SNE algorithm that favors the preservation of local trajectories.

Despite their legitimacy, these low dimensional vermiform trajectories hamper density-based segmentation techniques since they stand out very clearly from one another and from the background and will likely be labeled as individual clusters. Representative sampling of the prior to the embedding stage disbands the vermiform trajectories, yielding a more homogeneous and isotropic distribution of data points (Berman et al., 2014). Nonetheless, the available dataset, being composed of small individual recordings (900 frames long), does not suit the requirements for the described sampling step. Furthermore, video reconstruction would be equally frustrated since complete sequences of successive frames would rarely go through the dimensionality reduction step and, in order to complete the videos, missing frame labels would have to be speculated with the information provided by the labels of their nearest embedded neighbors. Alternatively, to preserve the video reconstruction step, segmentation parameter tuning or substitute dimensionality reduction techniques, such as PCA, can be deliberated to cope with the issue of embedded vermiform trajectories.

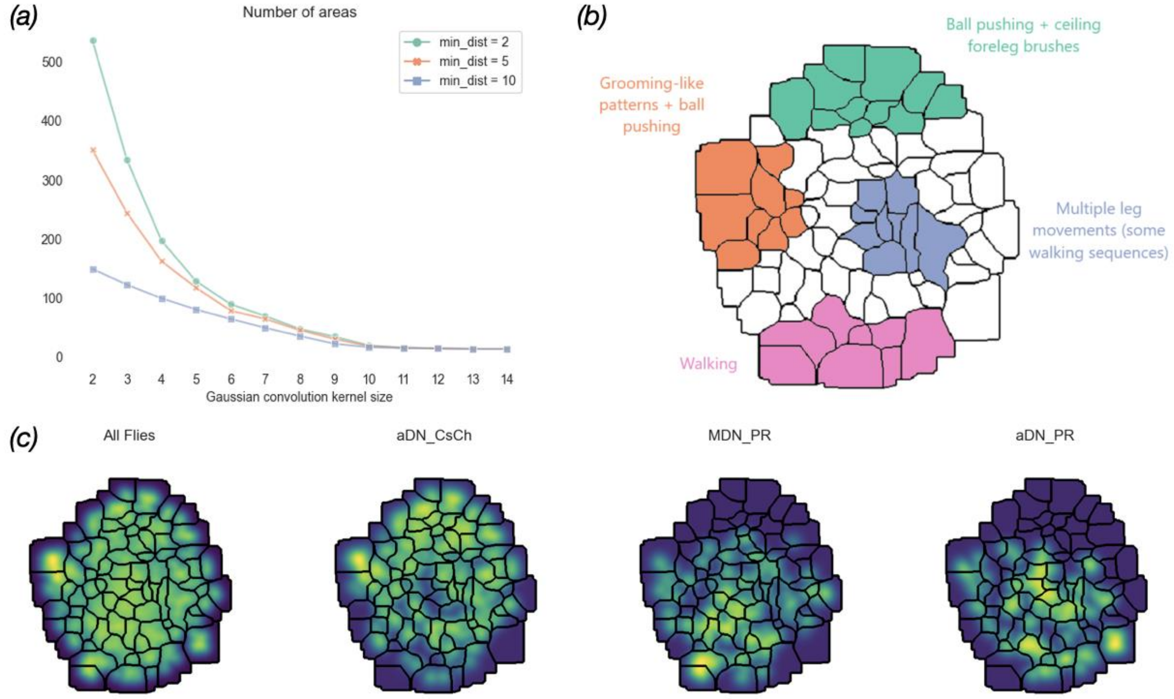


Figure 4.3: Clustering from *t-SNE*$_2$*-Watershed*. **(a)** Influence of key hand tuned parameters from Gaussian-kernel convolution (kernel size, σ) and Watershed clustering (*minimum distance*). **(b)** Organization of the behavioral space from visual inspection of cluster videos **(c)** Genotypic disparities in behavior: control *MDN-GAL4/+* and *aDN-GAL4/+* flies mostly perform forward walking, or actions with multiple limb involvements, while *aDN>CsChrimson* flies prefer to groom or interact with the stage's ceiling. Although *resting* frames were previously filtered, some resting sequences (mostly by *aDN>CsChrimson* flies) feature among the cluster videos.

Under default settings, the watershed algorithm yields ~80 areas, with diverse sizes and shapes, each containing a local maxima of the embedded plane's estimate PDF, $P(z_1,z_2)$. The number of areas can range from 542 to 14, depending on the tuning of Gaussian convolution kernel size and Watershed minimum distance (Figure 4.3(a)). The results are consistent due to the deterministic nature of the density estimation and segmentation steps.

Visual inspection of the videos that were reconstructed from the frames of each area (link in Section 3.8), under default parameters, shows that each trajectory within a cluster denotes a single behavior, although longer sequences that encompass more than one recognizable behavioral pattern are sometimes present. These prolonged video sequences are owed to the lengthy trajectories that populate the embedded plane and inconveniently bridge embedded points linked to different behaviors. To correct, or at least attenuate this effect, low frequency channel information can be removed from time-frequency analysis, allowing the disassemble of said long vermiform trajectories, at the expense of some insights towards the fly's slower, or faster but regular, movements. Supplementary data includes clustering results with minimum frequencies of 5, 7 and 10Hz that corroborate this effect.

Regarding cluster quality, although each cluster usually denotes a stand-out behavioral pattern, it is common to see clusters harbor more than one discernible behavior. The fact that, within a given area, the excessive behaviors are usually backed by more than a single behavioral bout suggests that this might occur as a consequence of two areas being merged together by the density estimation step - two or more close clusters could be blended together if the Gaussian convolution kernel size, σ, was too large. No additional efforts were made to further improve cluster quality through the decrease of σ due to time constraints, but finding the appropriate value of σ for a given dataset is encouraged as a measure of good practice when resorting to this density estimation technique. In parallel, it's noticeable that, due to occlusions that hamper the effectiveness of *DeepFly3D*'s landmark tracking network, a small portion of embedded sequences display no movements whatsoever - despite their constituent frames being labeled as *active* and the network reporting landmark movements - and are clustered along with genuine behavioral segments.

Most embeddings share a common global arrangement (often shifted by just a simple rotation operation) such as the one in Figure 4.3(b), where the upper left side of the embedded plane is dominated by ball pushing and/or spinning behaviors, with the upper end favoring simultaneous foreleg ceiling brushes and the left side endorsing grooming-like patterns, whereas the lower right side gradually shows less ball pushing and/or spinning in favor of behavior patterns that demand the involvement of all of the fly's legs, such as walking. Most embedded areas feature all three of the genotypes included in this dataset, apart from those situated at the top of the embedded plane, which are exclusive to *aDN>CsChrimson* individuals. Nonetheless, each individual fly recording seems to be confined to a relatively restricted region of the map, for the most part of its recording. Recordings stemming from the same fly are naturally closer to one another, as is the case with recordings from same genotype individuals (Figure 4.3(c)). Accordingly, the *fly homogeneity* metric has an average value of 35473 units, ranking 3rd among the employed pipelines (Table 4.1). *Cluster entropy* and *mean dwell time* metrics also feature among the top ranked values in their respective categories, with the main downgrades for this pipeline being the number of key hand tuned parameters (2 required by t-SNE and 4 by the segmentation step) and its run time (average of 21 minutes).

4.3 Clustering with complementary modular configurations

4.3.1 Dimensionality reduction with PCA

Due to the predicaments that arise when performing t-SNE over a full (not sampled) dataset that includes information from low frequency channel time frequency analysis, PCA is to be considered as a valid substitute embedding algorithm. As discussed in Section 2.4.1.1 , PCA aims to reduce the dimensionality of the posture-dynamics space while sacrificing local authenticity in favor of consistency at a global scale. Since neighboring frames in the posture-dynamics space are too much alike, PCA could prove itself more adequate for embedding such a dataset without the need for an intermediate sampling step, followed by multiple embeddings of individual fly datasets at a time, onto the same plane.

A first glance at the middle panel from Figure 4.5 suggests a more homogeneous embedding without discernible vermiform trajectories and segregation according to *frame activity* values. However, upon closer inspection, an overlap between the region of highest activity and some low activity trajectories becomes noticeable. Genotype-wise, the embedding seems to somewhat distinguish between *aDN>CsChrimson* and the two other control categories, with the former being globally represented across the embedded plane, but occupying mostly low activity regions and the *MDN-GAL4/+* and *aDN-GAL4/+* genotypes generally avoiding regions of lower activity (perpetuating the trend in Figure 4.3(c) - supplementary results are provided by the repository in Section 3.8).

Compared to *t-SNE$_2$-Watershed*, *PCA$_2$-Watershed*'s cluster *mean dwell time* decreases significantly from 25 to a mere 6 frames (below the outlier threshold for video reconstruction and the lowest value observed among the employed pipelines), implying that the trajectories race all across the embedded plane without pausing at particular locations that, otherwise, would translate to stereotypical behaviors. In parallel, *cluster entropy* dropped from 4.079 (*t-SNE$_2$-Watershed*) to 3.086 units (lowest ranked among all pipelines), indicating that cluster frames are less evenly distributed, i.e., a few clusters tend to dominate, which in turn means that fewer state transitions are to be expected and, hence, clustering will be less useful. Markov transition matrices and the probability mass functions of frames per cluster are portrayed in Figure 4.7 for *t-SNE$_2$-Watershed* and *PCA$_2$-Watershed* to illustrate this point.

Visual assessment of cluster videos also reveals substandard performance: cluster videos are frequently polluted with very brief (just over 10 frames long) resting behavioral sequences, with some sustaining tracking errors; globally, video durations are very short, in line with the *mean dwell time* drop under PCA, frustrating the validation of standout behaviors from visual inspection; many cluster videos are comprised of just a single behavioral sequence (a symptom of low *cluster entropy*) and ergo lack any cross-validation elements, rendering the clusters meaningless. Nevertheless, the few larger clusters are usually accurate, in particular when concerned with grooming behaviors.

Although PCA is substantially faster than its t-SNE contender (2615x faster under the current dataset, taking a mere 2.275 seconds), solves the vermiform-shaped embedded trajectories, is fairly unbiased and deterministic (hence, reproducible), cluster metrics and visual inspection of clustering results deem it unfit to embed a dataset with these particular characteristics while maintaining an adequate performance. Nonetheless, PCA could still be put to advantage if combined with t-SNE in such a way that would allow the user to find the appropriate middle ground between local and global data structure conservation. This translates to embedding the data onto *X* dimensions with PCA, followed by a definitive t-SNE embedding onto 2 dimensions. Among the examined values for *X*, *X=30* yielded the best results regarding cluster quality, while explaining approximately 85% of the dataset's variance (Figure 4.4). Embedded trajectories look visibly healthier - although vermiform trajectories are still present throughout the embedded plane (as expected) but are customarily quickly dissipated. Furthermore, activity partitioning in embedded data points became clearer with the hybrid *PCA$_X$-t-SNE$_2$* embeddings, *cluster entropy* increased (now ranks 2nd among all pipelines), *mean dwell times* become adequate for video reconstruction of the clustering results. This comes at a cost of loss of total reproducibility, although still better than *t-SNE$_2$-Watershed*, and increased running time (Table 4.1).

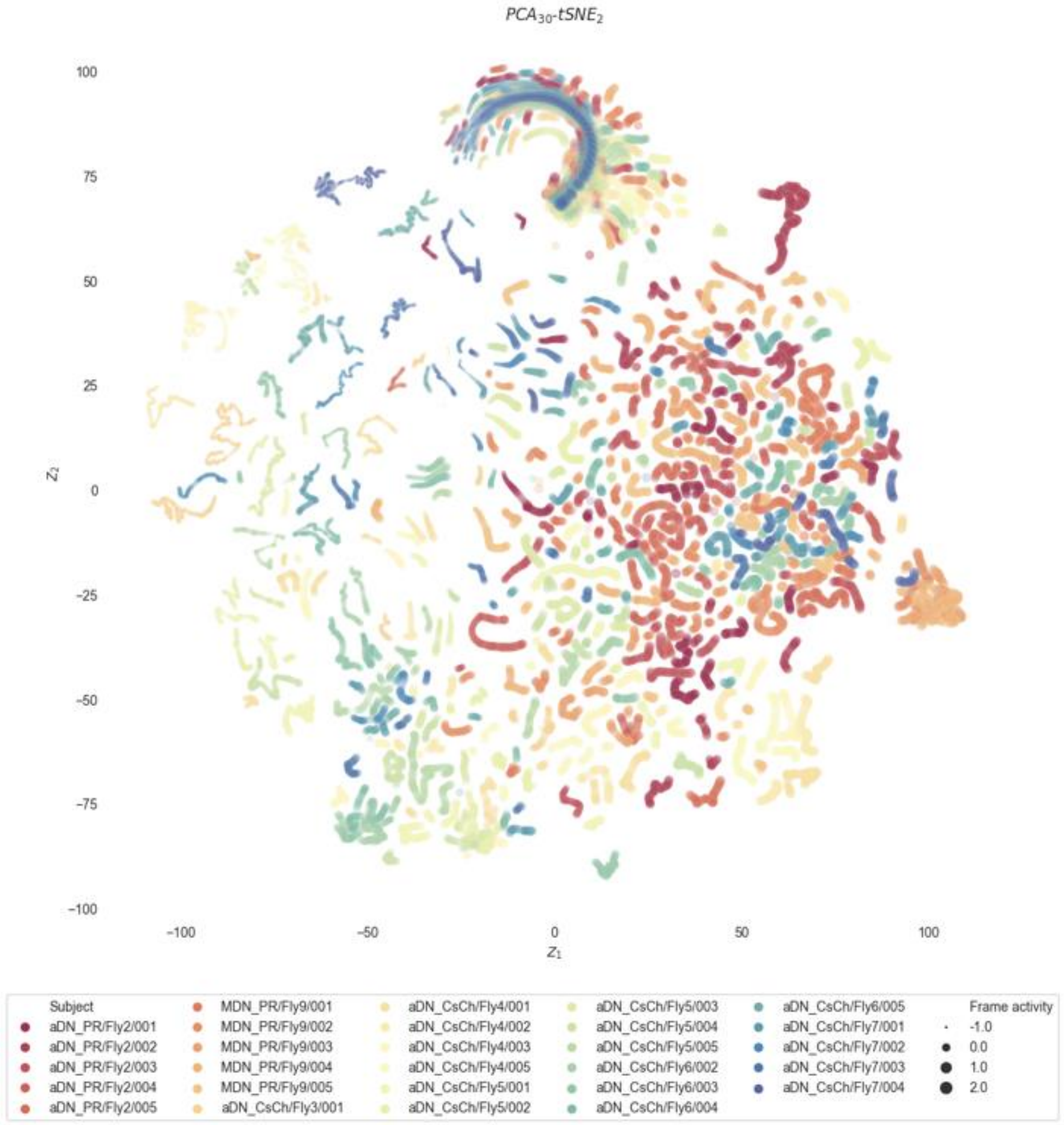


Figure 4.4: *PCA_{30}-t-SNE_2* embedding. Data points are colored by fly tag and their sizes are determined by their activity value (Section 3.3.5). Trajectories comprise only datapoints from the same fly and rarely intersect. The left side is dominated by low activity trajectories and features only *aDN>CsChrimson* flies. Again, *MDN-GAL4/+* and *aDN-GAL4/+* flies partake mostly in more energetic motions. The topmost region holds data points from the recording's edges and can easily be suppressed.

Regarding *fly homogeneity*, none of the embedding techniques completely succeeded in dissipating differences among genotypes, as same genotype fly centroids are globally embedded closer to one another (Figure 4.6). One should keep in mind that, on average, flies from different genetic lineages will tend to favor different behaviors which is reflected in their centroid coordinates. Nonetheless, Figure 4.6 reveals that *PCA_2* embeddings are somewhat more homogeneous than those by *t-SNE_2* and *PCA_{30}-t-SNE_2* since fly genotypes are less neatly distanced from one another. Data gathered

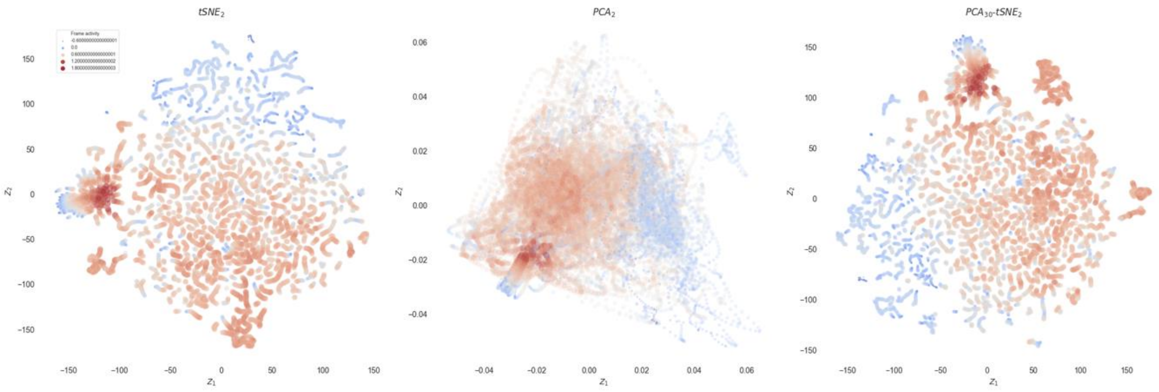


Figure 4.5: Embedding from each dimensionality reduction module. *t-SNE$_2$* on the left panel; *PCA$_2$* on the middle panel; *PCA$_{30}$-t-SNE$_2$* on the right panel. Cluster videos from *t-SNE$_2$* and *PCA$_{30}$-t-SNE$_2$* outshine those from *PCA$_2$* - which features more dispersed and overlapping trajectories.

in Table 4.1 corroborates this, with *PCA$_2$-Watershed* carrying a reasonable advantage over its concurrence in this domain.

4.3.2 Clustering with GMM-posterior probability assignment

Clustering with Gaussian convolution followed by a watershed transform relies heavily on the Gaussian kernel size parameter - when the value is too small, the density estimation step spawns an excessive number of local maxima from witch the Watershed algorithm floods the inverted PDF, yielding an excessive number of clusters; when too large, distant points in the embedded space are assimilated by large clusters with disregard to their behavioral nature, corrupting cluster quality.

Meanwhile, GMM relies on the assumption that the embedded points are drawn from the overlap of *k* multivariate Gaussian distributions called *mixture components*, bypassing the need for any additional hand-tuned parameters. Although the number of clusters (determined by the value of *k*) is established *a priori* by the user, the algorithm offers analytical benchmarks for clustering success: the *Bayesian Information Criterion* (BIC) and *Akaike Information Criterion* (AIC) (Section 3.4.2). Local

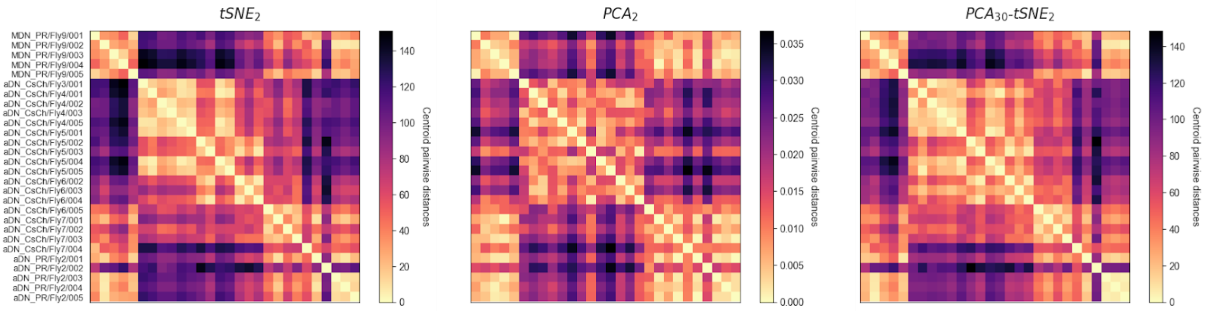


Figure 4.6: *Fly homogeneity* under each dimensionality reduction module. The distance matrices are computed from each fly's core coordinates to assess whether the embedding module has a significative influence over whether different flies are mapped closer or farther apart. Although there are noticeable contrasts between flies with different genotypes (different genotypes promote different behaviors), the considered embedding modules agree with each other. *PCA$_2$* is more successful at bridging different genotypes, although this is to be expected from its characteristic disperse and extensive trajectories.

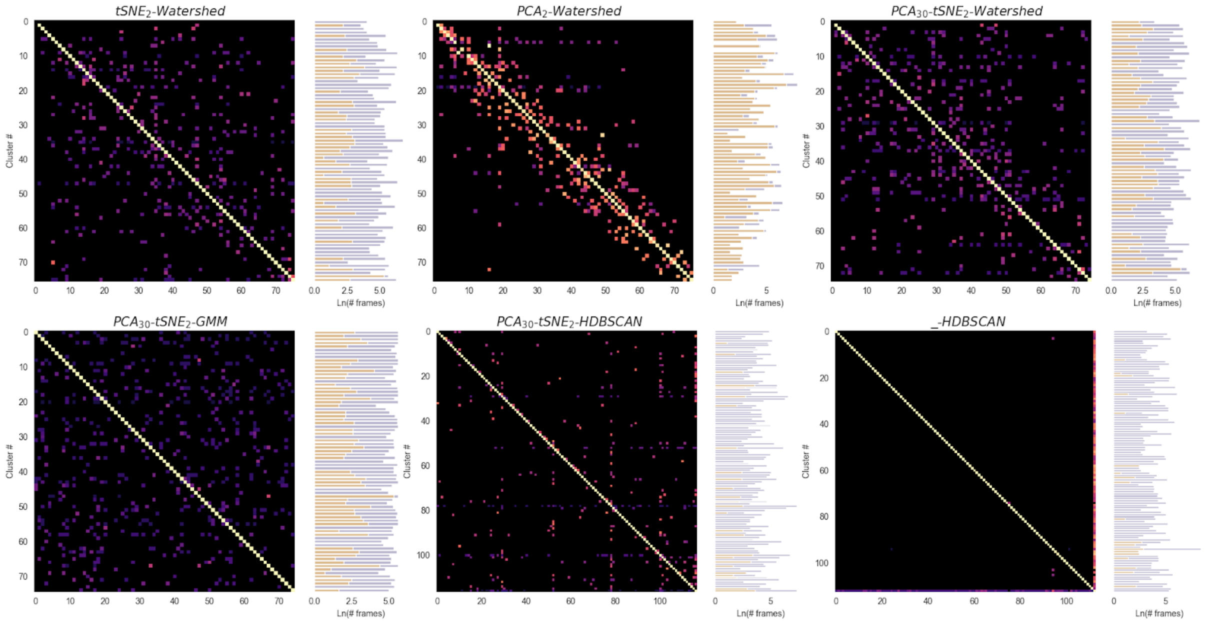


Figure 4.7: *Entropy* and cluster transitions. The panels hold Markov transition matrices, where each entry *{i,j}* is the probability of a data point transitioning from cluster *i* to cluster *j*, and histograms that reflect the logarithm of the number of frames per cluster (yellow indicates outlier frames, while purple expresses the total number of frames). *PCA*$_2$ results in a problematic number of outlier sequences. HDBSCAN promotes low *cluster* entropy values in both high- and low-dimensional spaces, which explains the odd Markov transition matrices (larger clusters dominate the transitions, while smaller clusters are visited very few times). Watershed and GMM based methods clustering provide more desirable clustering results.

minima of BIC and AIC can hint at the legitimate value of clusters in the embedded plane. However, BIC and AIC perpetually decrease along the *k* axis (Figure 4.8), suggesting that the number of clusters should surpass 150 (testing didn't go beyond *k=150* due to the low verisimilitude of a model contemplating so many clusters with this sort of dataset). The perpetual decrease of BIC and AIC suggests that, for GMM, the optimal fitting would consider each vermiform trajectory in the embedded space as a cluster, since these trajectories are quite compact and separated from one another by empty space. Otherwise, mixture component centers are likely stationed in such empty spaces and the vermiform trajectories placed under them are assigned lower posterior probabilities - the likelihood function decreases and the criteria BIC and AIC increase (equations 3.8 and 3.9). In line with the number of clusters generated by *t-SNE*$_2$*-Watershed* under default parameters, the value of *k* is set to 75 since it matches a local minimum in the BIC and AIC curves. GMM then yields a generative probabilistic model that describes the data distribution and clusters are assigned with its posterior probabilities.

Clusters in *PCA*$_{30}$*-t-SNE*$_2$*-GMM*'s embedded plane are rounder and more homogeneous in their size when compared to those in *PCA*$_{30}$*-t-SNE*$_2$*-Watershed* (Figure 4.9). Cluster videos from *PCA*$_{30}$*-t-SNE*$_2$*-GMM* were the top ranked in terms of video accuracy among all pipelines. Although still heavily polluted by mislabeled *resting* frames (where the joint labels sway meagerly, but with high frequency), most of its clusters share a standout behavioral bout, whether this bout is observed throughout the complete video sequences or is merely a common sight within them. Clusters portraying walking or grooming are usually more accurate and clusters depicting slow movements such as ball pushing, spinning or sluggish walking being more troublesome (supplementary data - *PCA*$_{30}$*-t-SNE*$_2$*-GMM* cluster videos). Another persisting issue is the fusing of two or more distinct behavioral bouts in some clusters, which comes as a consequence of large or misshapen mixture components this time.

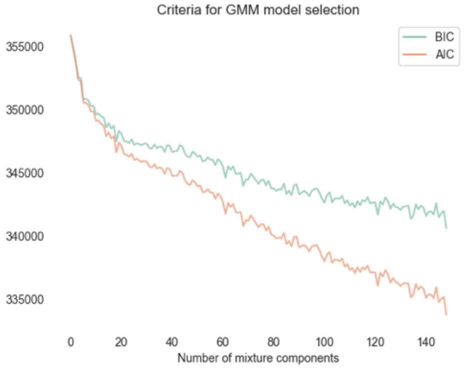


Figure 4.8: Determining the number of GMM mixture components. The ideal number of GMM mixture components is best assessed by finding the knee of the AIC and BIC curves (lower values suggest better fits). Here, it is suggested that this number should be prohibitively low (possibly implying that each trajectory should be deemed as a cluster).

Metric wise, *PCA*$_{30}$*-t-SNE*$_2$*-GMM* also yields better clusters, when compared to previous pipelines: *mean dwell time* is valued at 21.256 frames, which eases the visual inspection of clustering results, although lengthy video sequences that portray more than one discernable behavior are sometimes observed (splitting the embedded trajectories at the right instance seems to be a significant challenge for the embedding algorithms). The clusters size agreement is validated by an increase in *cluster entropy*, which, at 4.256 (98.6% of the reference value by the random cluster assignment), reached the maximum value among the examined pipelines. Figure 4.7 also indicates that *PCA*$_{30}$*-t-SNE*$_2$*-GMM* comprises fairly unpredictable cluster transitions as a consequence of its high *cluster entropy*, which can convey a more useful clustering. The number of key hand tuned parameters in *PCA*$_{30}$*-t-SNE*$_2$*-GMM* amounts to 4: the number of PCA modes to be kept, the t-SNE perplexity and distance metric, and the number of GMM's mixture components. The pipeline's run time is equally satisfactory, at roughly 14 minutes (Table 4.1).

4.3.3 Clustering of embedded data with HDBSCAN

The compelling argument in favor of HDBSCAN relies on the fact that it attempts to provide a cluster hierarchy, which might suit a possible underlying hierarchical organization of behavior, or, at least, help guide its exploration. Although HDBSCAN is a better match for morphologically asymmetric clusters than GMM, in the current settings it must overcome the heterogeneity of the embedded plane, composed of dense vermiform trajectories that contrast highly with the empty space that engulfs them.

A prominent feature of *PCA*$_{30}$*-t-SNE*$_2$*-HDBSCAN* is the disproportional size of its clusters (Figure 4.9). If two embedded vermiform trajectories are mapped very far from one another, but are linked by a chain of trajectories in between, with narrow gaps between its elements, HDBSCAN might still be able to cluster them together. This can lead to unsound clusters such as the one featured at the top of the bottom left panel from Figure 4.9 (colored purple), which is bridged to another cluster below; or clusters that exhibit implausible protuberances. Consequentially, enlarged clusters which frequently harbor multiple kinds of behavioral bouts are created, as depicted in the cluster videos from *PCA*$_{30}$*-t-SNE*$_2$*-HDBSCAN* (supplementary data - *PCA*$_{30}$*-t-SNE*$_2$*-HDBSCAN* cluster videos). The large clusters contrast with numerous small clusters, often composed of just a single behavioral bout sequence, from which no cluster quality assessments can be made. Moreover, just as with the previous clustering efforts, the cluster videos are pestered by resting sequences. Nonetheless, medium size clusters from *PCA*$_{30}$*-t-*

SNE$_2$-HDBSCAN were somewhat satisfactory, namely when concerned with walking or grooming patterns. The pipeline ranks 4th in cluster video quality, accordingly (Table 4.1).

Regarding other performance metrics, *PCA$_{30}$-t-SNE$_2$-HDBSCAN* keeps operating below the average standards set by the employed pipelines. *Cluster entropy*, at 3.616 units (ranked 4th), is lackluster, as expected given the unbalance in cluster dimensions. Unlike with *t-SNE$_2$-Watershed*, *PCA$_{30}$-t-SNE$_2$-Watershed* and *PCA$_{30}$-t-SNE$_2$-GMM*, the Markov transition matrix for *PCA$_{30}$-t-SNE$_2$-HDBSCAN* shows a more predictable pattern of cluster transitions, overshadowed by transitions to and from the dominant clusters, whereas minor clusters only feature self-transitions accompanied by one single arrival and departure transition (Figure 4.7). The pipeline's *mean dwell time* is set at an average 31 frames, which, in fairness, is still considered a plausible value, although it lies beyond the typical length (10-20 frames) of observed behavioral bouts from this dataset. Consistent with the pipelines that make use of both PCA and t-SNE in this scenario, *PCA$_{30}$-t-SNE$_2$-HDBSCAN* requires the hand tuning of at least 3 parameters at the dimensionality reduction stage, plus an added 3 at the HDBSCAN segmentation stage (*minimum cluster size*, *minimum samples* and the distance at which clusters are

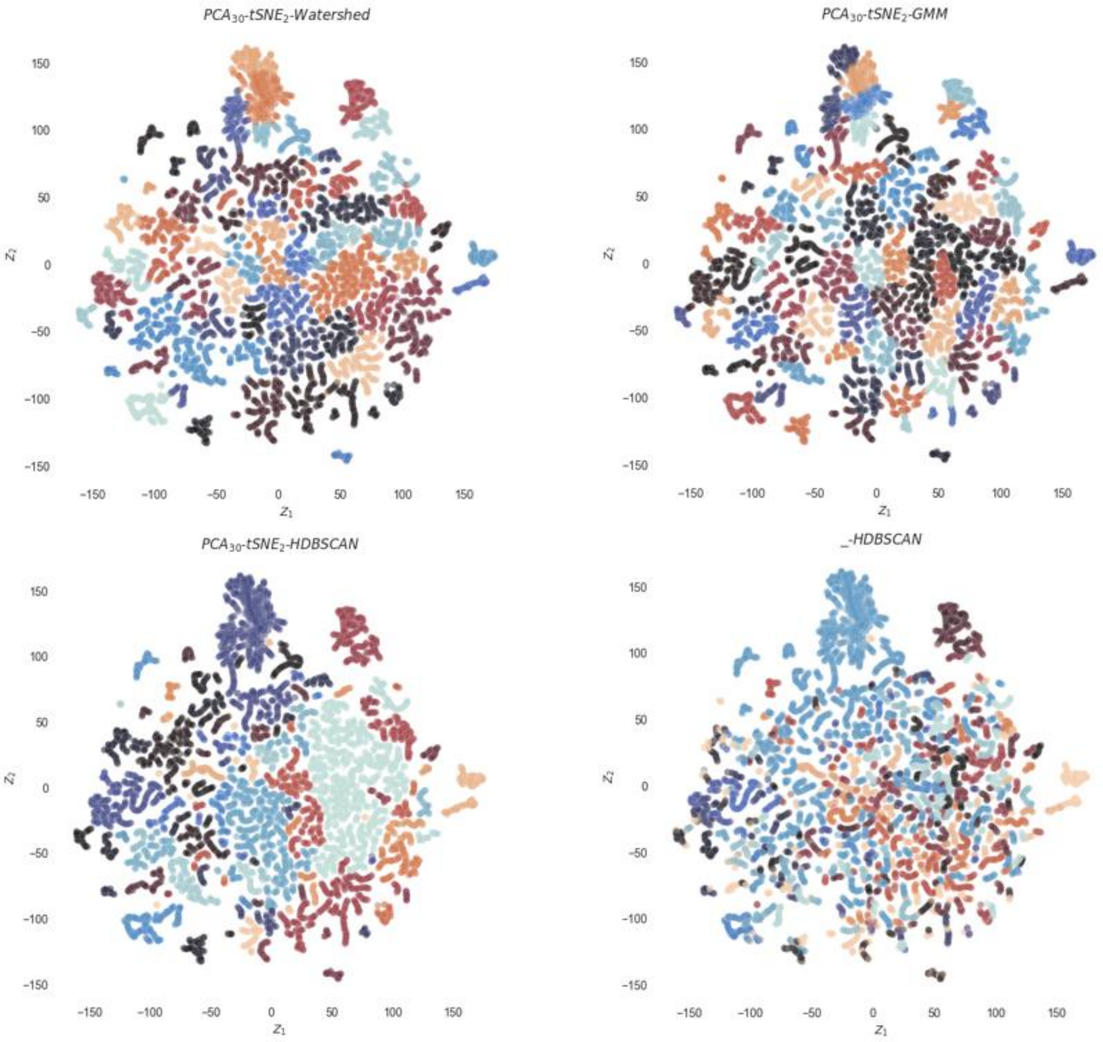
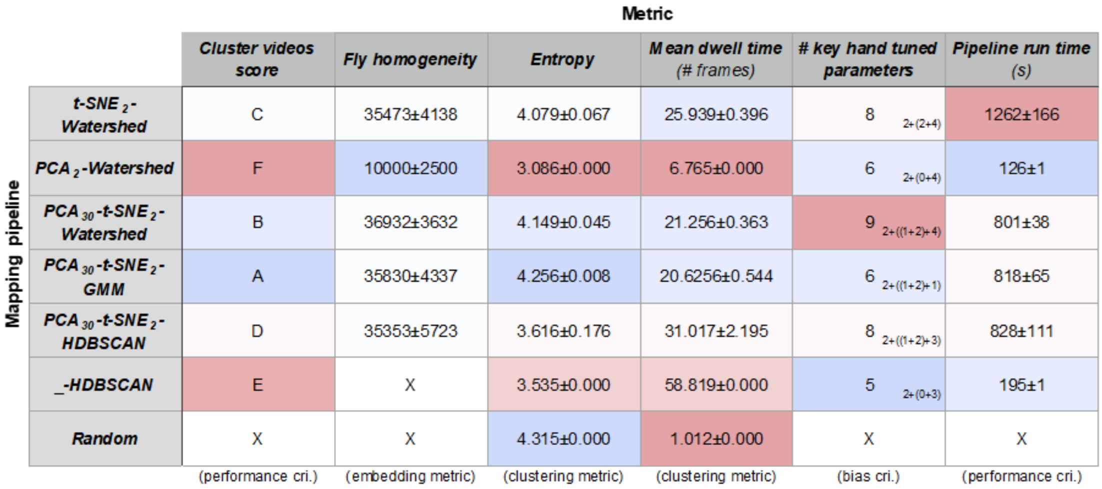


Figure 4.11: Comparing clustering modules. Once more, Watershed and GMM clustering yield more favorable results: clusters are appropriate-sized (GMM clusters are rounder, given they arise from 2D Gaussian distributions). HDBSCAN clusters are disproportional in size and can link very distant trajectories (polluting clusters with disparate bouts of behavior). Furthermore, the bottom panels reveal a significant discrepancy between high- and low-dimensional clustering.

Table 4.1: Pipeline metric scores. Each pipeline was run a total of 8 times. The colors indicate the degree of metric desirability (with blue being most desirable and red the least). When a given metric is inapplicable, the entry is marked by an *X*.



Mapping pipeline	Metric					
	Cluster videos score	*Fly homogeneity*	*Entropy*	*Mean dwell time (# frames)*	*# key hand tuned parameters*	*Pipeline run time (s)*
t-SNE$_2$-Watershed	C	35473±4138	4.079±0.067	25.939±0.396	8 $_{2+(2+4)}$	1262±166
PCA$_2$-Watershed	F	10000±2500	3.086±0.000	6.765±0.000	6 $_{2+(0+4)}$	126±1
PCA$_{30}$-t-SNE$_2$-Watershed	B	36932±3632	4.149±0.045	21.256±0.363	9 $_{2+((1+2)+4)}$	801±38
PCA$_{30}$-t-SNE$_2$-GMM	A	35830±4337	4.256±0.008	20.6256±0.544	6 $_{2+((1+2)+1)}$	818±65
PCA$_{30}$-t-SNE$_2$-HDBSCAN	D	35353±5723	3.616±0.176	31.017±2.195	8 $_{2+((1+2)+3)}$	828±111
_-HDBSCAN	E	X	3.535±0.000	58.819±0.000	5 $_{2+(0+3)}$	195±1
Random	X	X	4.315±0.000	1.012±0.000	X	X
	(performance cri.)	(embedding metric)	(clustering metric)	(clustering metric)	(bias cri.)	(performance cri.)

collected from the single linkage tree, *λ*). This pipeline shares a similar run time with the remaining *PCA$_{30}$-t-SNE$_2$-_* pipelines (~14 minutes), given that the time bottleneck occurs the dimensionality reduction step.

4.3.4 Clustering of high-dimensional data with HDBSCAN

Given the *curse of dimensionality*, which hampers the success of density based clustering approaches in high dimensional spaces due to the rapid increase of volume with the number of dimensions, accompanied by an increase in data sparsity, the *_-HDBSCAN* pipeline was met with some disbelief from the start. Nonetheless, its conception was motivated its simplicity and curiosity over whether there is a hierarchical structure that governs the flies' posture-dynamics space that might substantiate a hierarchical nature in the flies' behavioral organization, and, also, if such a hierarchical structure is conserved once the posture-dynamics space expression matrix is embedded with t-SNE or PCA.

To ease computational requirements while maintaining a truthful data structure, the data is embedded onto a 125 dimensions manifold with PCA, which is sufficient to explain >95% of variance in the data. Only then HDBSCAN clustering is performed. In parallel, HDBSCAN was employed over the same *t-SNE$_2$-PCA$_{30}$* embedding depicted in Figure 4.9 - whose results were scrutinized in the previous sections. As anticipated, high dimensional clusters are not in agreement with the embedded ones, despite sharing a few overall features, such as the disproportion in cluster sizes and the inflated number of single trajectory clusters. Accordingly, *t-SNE$_2$-PCA$_{30}$-HDBSCAN* and *_-HDBSCAN* possess distinct cluster hierarchical structures: while the main cluster in the former pipeline eventually breaks off at *λ~7*, yielding several clusters of similar size that ultimately break off themselves; the main cluster in *_-HDBSCAN* is preserved throughout the descent of *λ*, having only a few small clusters breaking from it at a time (Figure 4.10(a)). High dimensional clustering also tends to come with lower scores regarding outliers (Figure 4.10(b) - higher score means the point is more likely to be an outlier).

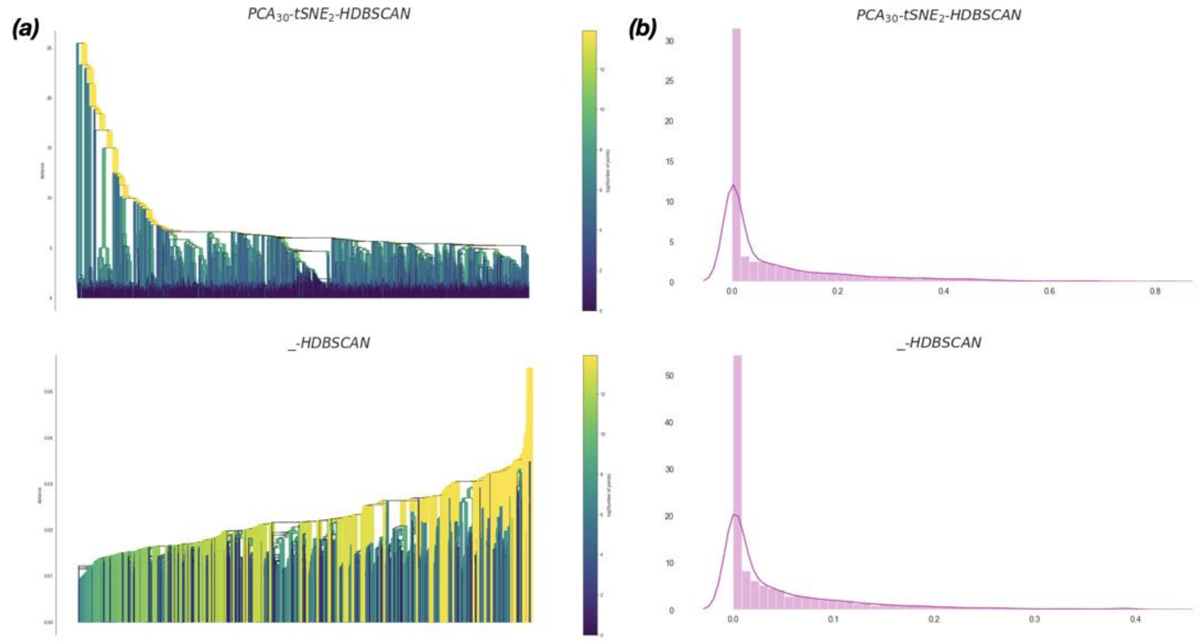


Figure 4.13: Hierarchical structure of high- and low-dimensional spaces. **(a)** HDBSCAN *single linkage tree* from the posture-dynamics space (bottom) and its embedded rendition (top). The low-dimensional space shows a clearer hierarchical structure, albeit its absence in the original high-dimensional space. **(b)** Outlier cluster scores from low- (top) and high-dimensional (bottom) spaces – in the later, there is an overall greater tendency for a given point to be considered as an outlier.

Cluster videos, namely from medium sized clusters, show some consistency, although pollution from resting videos persists globally. Oddly, most behavioral sequences come from the first recording of *aDN_PR_Fly2*, with other flies likely having most of their frames being labeled as outliers due to their sparsity or being labeled together in a very large supercluster. Cluster metrics suggest substandard values for *entropy* (as expected, given the disproportionality of cluster sizes) and *mean dwell time* (larger clusters increase the chances of consecutive trajectories falling on the same cluster and being appended into a larger behavioral sequence, if by any chance that's not the case to begin with), both ranking 5th among the tested pipelines. The odd cluster configuration of *_-HDBSCAN* leads to an off-putting Markov transition matrix, where cluster transitions occur either from the cluster to itself or from the cluster to the cluster with the highest label, with very few exceptions to this rule (Figure 4.7). The remaining criteria for number of key hand tuned parameters and run time are amongst the best, with only 3 key hand tuned parameters (all regarding HDBSCAN) and an average run time of 3.25 minutes, albeit these advantages are meager when considering the shortcomings in the remaining categories.

Chapter 5

Conclusions

The aim of this research was to achieve unsupervised quantification of behaviors by *D. Melanogaster* under minimally constrained conditions, while relying on three postulated principles of behavior: *low postural dimensionality*, *stereotypy* and *discretization*; with an additional contribution by *hierarchical organization*. A computational framework was set up to hold combinations of versatile modules that cooperatively accomplish behavioral classification, as well as unsupervised benchmarks for their success. Among the candidate pipelines, *PCA_{30}-t-SNE_2-GMM* yielded the best overall results, delivering globally consistent clusters and manifesting positive metric scores. Though it relies substantially on the values of 6 hand tuned parameters, 4 of them (t-SNE *perplexity* and distance metric, PCA dimensions and number of GMM mixture components) can be computed through ingenious, unbiased procedures - bypassing the need for user specific intuitions. Likewise, unbiased evaluation of candidate pipeline performances was carried out by invoking 3 impartial and meaningful metrics (*fly homogeneity*, *entropy* and *mean dwell time*), as well as other two performance criteria (number of key hand tuned parameters and pipeline run time), that aim to complete a cycle of full automation from the conception of a definition of behavior to the attestation of the results yielded by such definition.

The methodology, although not groundbreaking in its entirety, made use of an unprecedented tool for accurately tracking the 3D pose in animals as small as *D. Melanogaster*, *DeepFly3D*, which paved the way for novel feature angle normalization across individual flies and, potentially, across species of comparable insects. Although the analysis of feature angle normalization's significance was severely cut short by the implementation of an activity filter in the preprocessing stage, which hampered a meaningful interpretation of these results, it remains a conceptually powerful asset when dealing with a pose dataset from morphologically disparate individuals.

Unexpectedly, embeddings by *t-SNE_2* or *PCA_X-t-SNE_2* were profoundly impacted by the admittance of low frequencies (between 1 and 3Hz) in the time-frequency analysis stage, which culminated with the presence of impractical, long vermiform trajectories across the embedded space. The low frequency oriented wavelet scales yield slow changing wavelet transform coefficients that are included in the rows of the posture-dynamics expression matrix - consequently, the distances between columns of this matrix are shortened, which is detected by meticulous embedding algorithms such as t-SNE. Embedded vermiform trajectories become, on average, longer when lower minimum frequencies are considered, bridging further instances of behavior that ought to be embedded separately. Nonetheless, solutions are available - for this dataset, minimum frequencies above 3Hz yield more truthful trajectories with sensible dwell times, given the average length of the flies' typical behavioral bouts. An alternate solution involves performing a sampling operation prior to the dimensionality reduction step, as to partition the links created by the slow varying coefficients. However, this requires performing multiple embeddings per run, prohibits consistent labeling across all data points and needs additional parameters to be adjusted according to user specific intuitions.

Among the candidate techniques for clustering, the already proven Watershed and Gaussian Mixture Models outperformed the newfangled HDBSCAN. Although Watershed can provide satisfactory cluster videos, this technique relies heavily on user intuitions, rendering it less reproducible. To the contrary, GMMs require a single methodological choice parameter - for which unbiased selection criteria such as BIC and AIC are applicable - supplemented by less predictable clustering and more comprehensible cluster videos. Lastly, HDBSCAN struggled both in high and low dimensional spaces culminating with clusters of disproportional sizes (synonym of low cluster entropies). Clustering directly from the posture-dynamics expression matrix reveals a lack of hierarchic structure in this data space, since smaller clusters gradually fall off from one large cluster, which conserves its core throughout the extent of the algorithm. The embedded spaces that followed *t-SNE*$_2$ or *PCA*$_X$*-t-SNE*$_2$ displayed a more evident hierarchical arrangement, but nonetheless produced impractically unbalanced cluster sizes.

Unfortunately, the tested pipelines were far from delivering consensual results that could mutually back their rationales without the need for external benchmarks of their success. Embedding algorithms are thorny subjects due to their sensitivity to parameter values and arduous interpretations since they are known to occasionally create fake clusters (there is no way of knowing *a priori* whether the features discarded by the algorithm match noisy or true features). To increase confidence in the *clusterer*, the conception of a test set with *a priori* classification, either from different consensus clustering algorithms (unbiased) or manual (risk of user bias), is advised. Further research should include, as well: a broader comparison of embedding and/or clustering methods, particularly high dimensional clustering techniques (e.g. UMAP, Autoencoder for embedding and Subspace or Correlation Clustering in high dimensional spaces); sensitivity analysis to noisy subject pose data (a significant issue with resting labels from the current dataset); lastly, a more complete and revised set of unbiased performance metrics, with possible assessment of similarity between cluster video elements.

Bibliography

Bender, J. A., Simpson, E. M., & Ritzmann, R. E. (2010). Computer-assisted 3D kinematic analysis of all leg joints in walking insects. *PLoS ONE*. https://doi.org/10.1371/journal.pone.0013617

Berman, G. J. (2018). Measuring behavior across scales. In *BMC Biology*. https://doi.org/10.1186/s12915-018-0494-7

Berman, G. J., Bialek, W., & Shaevitz, J. W. (2016). Predictability and hierarchy in Drosophila behavior. *Proceedings of the National Academy of Sciences of the United States of America*. https://doi.org/10.1073/pnas.1607601113

Berman, G. J., Choi, D. M., Bialek, W., & Shaevitz, J. W. (2014). Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of the Royal Society Interface*. https://doi.org/10.1098/rsif.2014.0672

Bongard, J., & Lipson, H. (2007). Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*. https://doi.org/10.1073/pnas.0609476104

Breed, M. D., & Sanchez, L. (2010). Both Environment and Genetic Makeup Influence Behavior. *Nature Education Knowledge*.

Broekmans, O. D., Rodgers, J. B., Ryu, W. S., & Stephens, G. J. (2016). Resolving coiled shapes reveals new reorientation behaviors in C. elegans. *ELife*. https://doi.org/10.7554/eLife.17227

Brown, A. E. X., & de Bivort, B. (2018). Ethology as a physical science. *Nature Physics*, *14*(7), 653–657. https://doi.org/10.1038/s41567-018-0093-0

Brown, A. E. X., Yemini, E. I., Grundy, L. J., Jucikas, T., & Schafer, W. R. (2013). A dictionary of behavioral motifs reveals clusters of genes affecting Caenorhabditis elegans locomotion. *Proceedings of the National Academy of Sciences of the United States of America*. https://doi.org/10.1073/pnas.1211447110

Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-37456-2_14

Casiez, G., Roussel, N., & Vogel, D. (2012). 1€ filter: A simple speed-based low-pass filter for noisy input in interactive systems. *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/2207676.2208639

Daniels, B. C., & Nemenman, I. (2015). Automated adaptive inference of phenomenological dynamical models. *Nature Communications*. https://doi.org/10.1038/ncomms9133

Dankert, H., Wang, L., Hoopfer, E. D., Anderson, D. J., & Perona, P. (2009). Automated monitoring and analysis of social behavior in Drosophila. *Nature Methods*. https://doi.org/10.1038/nmeth.1310

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*.

Farruggia, M. C., & Small, D. M. (2019). Effects of adiposity and metabolic dysfunction on cognition: A review. In *Physiology and Behavior*. https://doi.org/10.1016/j.physbeh.2019.112578

Griffel, D. H., & Daubechies, I. (1995). Ten Lectures on Wavelets. *The Mathematical Gazette*. https://doi.org/10.2307/3620105

Günel, S., Rhodin, H., Morales, D., Campagnolo, J., Ramdya, P., & Fua, P. (2019). Deepfly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult Drosophila. *ELife*, *8*. https://doi.org/10.7554/eLife.48571

Jollife, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. In *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. https://doi.org/10.1098/rsta.2015.0202

Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S., & Branson, K. (2013). JAABA: Interactive machine learning for automatic annotation of animal behavior. *Nature Methods*. https://doi.org/10.1038/nmeth.2281

Katsov, A. Y., Freifeld, L., Horowitz, M., Kuehn, S., & Clandinin, T. R. (2017). Dynamic structure of locomotor behavior in walking fruit flies. *ELife*. https://doi.org/10.7554/eLife.26410

Kim, C. K., Adhikari, A., & Deisseroth, K. (2017). Integration of optogenetics with complementary methodologies in systems neuroscience. In *Nature Reviews Neuroscience*. https://doi.org/10.1038/nrn.2017.15

Lucas, C., & Sokolowski, M. B. (2009). Molecular basis for changes in behavioral state in ant social behaviors. *Proceedings of the National Academy of Sciences of the United States of America*. https://doi.org/10.1073/pnas.0809463106

Luo, L., Gershow, M., Rosenzweig, M., Kang, K. J., Fang-Yen, C., Garrity, P. A., & Samuel, A. D. T. (2010). Navigational decision making in Drosophila thermotaxis. *Journal of Neuroscience*. https://doi.org/10.1523/JNEUROSCI.4090-09.2010

Marques, J. C., Lackner, S., Félix, R., & Orger, M. B. (2018). Structure of the Zebrafish Locomotor Repertoire Revealed with Unsupervised Behavioral Clustering. *Current Biology*. https://doi.org/10.1016/j.cub.2017.12.002

Martinez, J., Hossain, R., Romero, J., & Little, J. J. (2017). A Simple Yet Effective Baseline for 3d Human Pose Estimation. *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV.2017.288

Marwala, T. (2018). Gaussian Mixture Models. In *Handbook of Machine Learning*. https://doi.org/10.1142/9789813271234_0013

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*. https://doi.org/10.1038/s41593-018-0209-y

Mendes, C. S., Bartos, I., Akay, T., Márka, S., & Mann, R. S. (2013). Quantification of gait parameters in freely walking wild type and sensory deprived Drosophila melanogaster. *ELife*. https://doi.org/10.7554/eLife.00231

Meyer, F. (1994). Topographic distance and watershed lines. *Signal Processing*. https://doi.org/10.1016/0165-1684(94)90060-4

Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M., & Mathis, M. W. (2019). Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nature Protocols*. https://doi.org/10.1038/s41596-019-0176-0

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-319-46484-8_29

Otsu, & N. (1996). A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man and Cybernetics*.

Pelleg, D., & Moore, A. (1999). *Accelerating exact k -means algorithms with geometric reasoning* . https://doi.org/10.1145/312129.312248

Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S. H., Murthy, M., & Shaevitz, J. W. (2019). Fast animal pose estimation using deep neural networks. *Nature Methods*. https://doi.org/10.1038/s41592-018-0234-5

Pérez-Escudero, A., Vicente-Page, J., Hinz, R. C., Arganda, S., & De Polavieja, G. G. (2014). IdTracker: Tracking individuals in a group by automatic identification of unmarked animals. *Nature Methods*. https://doi.org/10.1038/nmeth.2994

Richard, & Dawkins, M. (1976). Hierachical organization and postural facilitation: Rules for grooming in flies. *Animal Behaviour*. https://doi.org/10.1016/S0003-3472(76)80003-6

Seeds, A. M., Ravbar, P., Chung, P., Hampel, S., Midgley, F. M., Mensh, B. D., & Simpson, J. H. (2014). A suppression hierarchy among competing motor programs drives sequential grooming in Drosophila. *ELife*. https://doi.org/10.7554/eLife.02951

Shoji, H., Hagihara, H., Takao, K., Hattori, S., & Miyakawa, T. (2012). T-maze forced alternation and left-right discrimination tasks for assessing working and reference memory in mice. *Journal of Visualized Experiments*. https://doi.org/10.3791/3300

Stephens, G. J., Johnson-Kerner, B., Bialek, W., & Ryu, W. S. (2008). Dimensionality and dynamics in the behavior of C. elegans. *PLoS Computational Biology*.

https://doi.org/10.1371/journal.pcbi.1000028

Todd, J. G., Kain, J. S., & De Bivort, B. L. (2017). Systematic exploration of unsupervised methods for mapping behavior. *Physical Biology*. https://doi.org/10.1088/1478-3975/14/1/015002

van den Heuvel, M. P., & Sporns, O. (2019). A cross-disorder connectome landscape of brain dysconnectivity. In *Nature Reviews Neuroscience*. https://doi.org/10.1038/s41583-019-0177-6

Van Der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*.

Vogelstein, J. T., Park, Y., Ohyama, T., Kerr, R. A., Truman, J. W., Priebe, C. E., & Zlatic, M. (2014). Discovery of brainwide neural-behavioral maps via multiscale unsupervised structure learning. *Science*. https://doi.org/10.1126/science.1250298

Wiltschko, A. B., Johnson, M. J., Iurilli, G., Peterson, R. E., Katon, J. M., Pashkovski, S. L., Abraira, V. E., Adams, R. P., & Datta, S. R. (2015). Mapping Sub-Second Structure in Mouse Behavior. *Neuron*. https://doi.org/10.1016/j.neuron.2015.11.031

Wu, Y., Zhang, W., Li, J., & Zhang, Y. (2013). Optical imaging of tumor microenvironment. *American Journal of Nuclear Medicine and Molecular Imaging*.