A Tangible User Interface for Interactive Data Visualisation

by

Shiroq Al-Megren

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

The University of Leeds

School of Computing

January 2016

The candidate confirms that the work submitted is her own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

The work in Chapter 7 of the thesis has appeared in publication as follows:

Al-Megren, S. and Ruddle, R. A. Comparing tangible and multi-touch interaction for interactive data visualization tasks. In *Proceedings of the 10th International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2016), TEI '16, ACM, pp. 279-286.

Acknowledgements

I would like to take this opportunity to extend my thanks to all the people who helped and supported me throughout this PhD. It has been a long journey, but I have enjoyed it all.

First, I would like to thank my supervisor, professor Roy Ruddle for his guidance and encouragement throughout this research. Thank you for providing me with a valuable source of knowledge.

I would also like thank all the people, past and present, that occupied room 6.01. Thank you for all the wothwhile discussions, support and encouragement. You have been a great help, and it was a pleasure working besides you all.

Thanks go to the administration and support staff in the School of Computing and the University of Leeds. I am also grateful to all volunteers who took part in my studies.

Finally, I would like to show my gratitude to my family, and specially my parents, for their endless encouragement and support. I would also like to thank my friends who are scattered around the globe, but never failed to show their encouragement.

Abstract

Information visualisation (infovis) tools are integral for the analysis of large abstract data, where interactive processes are adopted to explore data, investigate hypotheses and detect patterns. New technologies exist beyond post-windows, icons, menus and pointing (WIMP), such as tangible user interfaces (TUIs). TUIs expand on the affordance of physical objects and surfaces to better exploit motor and perceptual abilities and allow for the direct manipulation of data.

TUIs have rarely been studied in the field of infovis. The overall aim of this thesis is to design, develop and evaluate a TUI for infovis, using expression quantitative trait loci (eQTL) as a case study. The research began with eliciting eQTL analysis requirements that identified high-level tasks and themes for quantitative genetic and eQTL that were explored in a graphical prototype.

The main contributions of this thesis are as follows. First, a rich set of interface design options for touch and an interactive surface with exclusively tangible objects were explored for the infovis case study. This work includes characterising touch and tangible interactions to understand how best to use them at various levels of metaphoric representation and embodiment. These design were then compared to identify a set of options for a TUI that exploits the advantages of touch and tangible interaction.

Existing research shows computer vision commonly utilised as the TUI technology of choice. This thesis contributes a rigorous technical evaluation of another promising technology, micro-controllers and sensors, as well as computer vision. However the findings showed that

some sensors used with micro-controllers are lacking in capability, so computer vision was adopted for the development of the TUI.

The majority of TUIs for infovis are presented as technical developments or design case studies, but lack formal evaluation. The last contribution of this thesis is a quantitative and qualitative comparison of the TUI and touch UI for the infovis case study. Participants adopted more effective strategies to explore patterns and performed fewer unnecessary analyses with the TUI, which led to significantly faster performance. Contrary to common belief bimanual interactions were infrequently used for both interfaces, while epistemic actions were strongly promoted for the TUI and contributed to participants' efficient exploration strategies.

Contents

List of Figures

List of Tables

Abbreviations

dbSNP	Single nucleotide polymorphism database
DNA	Deoxyribonucleic acid
eQTL	Expression quantitative trait loci
GUI	Graphical user interface
GWAS	Genome wide association study
HCI	Human computer interaction
Infovis	Information visualisation
MBV	Marker-based vision
PharmGKB	The pharmacogenomics knowledge base
QTL	Quantitative trait loci
RFID	Radio frequency identification
RNA	Ribonucleic acid
RSID	Registered serial number
Scivis	Scientific visualisation
SNP	Single nucleotide polymorphism
TAC	Token and constraints
TUI	Tangible user interface
UI	User interface
USE	Usefulness, satisfaction and ease of use
WIMP	Windows, icons, menus and pointing
WPF	Windows presentation foundation

Chapter 1

Introduction

1.1 Introduction

Tangible user interfaces (TUIs) and interactions are part of the recent innovation wave in human computer interaction (HCI) research that transcends graphical user interfaces (GUI) and windows, icons, menus and pointing (WIMP) interaction styles. While GUIs represent digital information as icons on a computer screen, TUIs map GUI representations to physical icons that act as containers or handles for the underlying digital information [105]. A TUI draws from the users' rich environment and augments it to respond to and address their needs. It utilises the users' existing skills and knowledge of interaction and thus reduces their mental workload and affect their reasoning ability and performance.

The novelty of TUIs as an interface style has sparked interest in various application fields, such as learning, entertainment, problem solving and social communication, where the data is limited in scale. This is not always the case in visualisation, where data grows in scale and interdependencies become intricate. Visualisation has roots in history that go back to the ancient Greek's alchemical cosmology [65], the Egyptian's turin papyrus visualising a geological map [89] and the Chinese's astronomy visualisations [45]. Visualisations are valuable tools for exploring information as they leverage a person's cognition and visual thinking. TUIs have the potential to further expand the benefits of visualisations by leveraging users' motor and perceptual abilities.

Data visualisation is commonly used as an umbrella term that covers scientific and information visualisation. Scientific visualisation (scivis) refers to the realistic representation of three dimensional phenomena and the graphic display of spatial data. Concepts in Information visualisation (infovis) are often abstract and are examined for visual metaphors. The two fields of research visualise large scale data sets from scientific processes that are inherently spatial (scivis) or non-spatial and highly dimensional data (infovis). TUI research has been carried out to physically visualise and control scientific processes such as the structure of molecules and bonding behaviour [75, 205] Compared to scivis, there has been little research into the use of TUIs for infovis which can clearly benefit from more direct and flexible interfaces.

In the present thesis, expression quantitative trait loci (eQTL) studies were used to guide the development and evaluation of a TUI for infovis. eQTL studies have become a popular research area in the field of genomics with ties to genome wide association studies (GWAS) of human diseases and pharmacogenomics. Growth in biological information, including data used in eQTL studies, owing to technological advances in the genomics field necessitates a change in interactive visualisation tools to better handle the size of the data, interdependencies and interactions. Despite the importance of eQTL studies, the tools provided for visualisation are limited in capability.

1.2 Research Questions

To evaluate how people might benefit from tangible interaction when interactively analysing abstract visualisations, i.e. infovis, the research described in this thesis focuses on answering the following questions:

What is the analysis workflow for an interactive infovis case study? After identifying generalisable high-level tasks for interactive infovis, the research explored novel tasks and functionalities on a graphical implementation. The effectiveness of touch and mouse inputs for combining eQTL results was also investigated.

How should a TUI for interactive infovis be designed? The research proposed a series of design options for an interactive surface TUI for infovis that balances between touch and tangible interactions. This was achieved with the aid of the token and constraints (TAC) framework [212] after exploring the design options for an exclusively tangible TUI at various levels of the tangible objects' specificity. These designs were balanced against a baseline touch user interface (UI) to overcome the limitations of touch and tangible interactions.

What technologies are suitable for infovis TUIs? The research investigated the technical feasibility of two prominent development technologies, microcontrollers' sensors and computer vision, for the infovis case study's sensing modalities: position, orientation and stack. The results of the investigations found that accelerometers, gyroscopes and force sensors were suitable for detecting short displacements, two-object stacks and wide-angle orientations. Whereby computer vision accurately detected position and orientation, but stacking required additional technologies. Based on these results, the research iteratively developed a TUI for interactive infovis which combines tangible and touch interactions. The system consisted of an interactive surface, case study visualisation application and tangible objects.

How effective is an infovis TUI compared with a touch UI baseline? The thesis provided evidence of the strengths of the infovis TUI compared to a touch UI baseline. The results showed that the adoption of tangible interactions helped users adopt a more effective strategy for detecting patterns across files and perform fewer unnecessary analyses. The results also showed that this is likely owing to the adoption of epistemic actions which were promoted by the tangible objects.

1.3 Thesis Outline

The thesis is organised as follows. **Chapter 2** reviews previous research, and **Chapter 3** provides an overview of quantitative genetics, which examines the relationship between genomes and traits. The thesis introduces the human genome

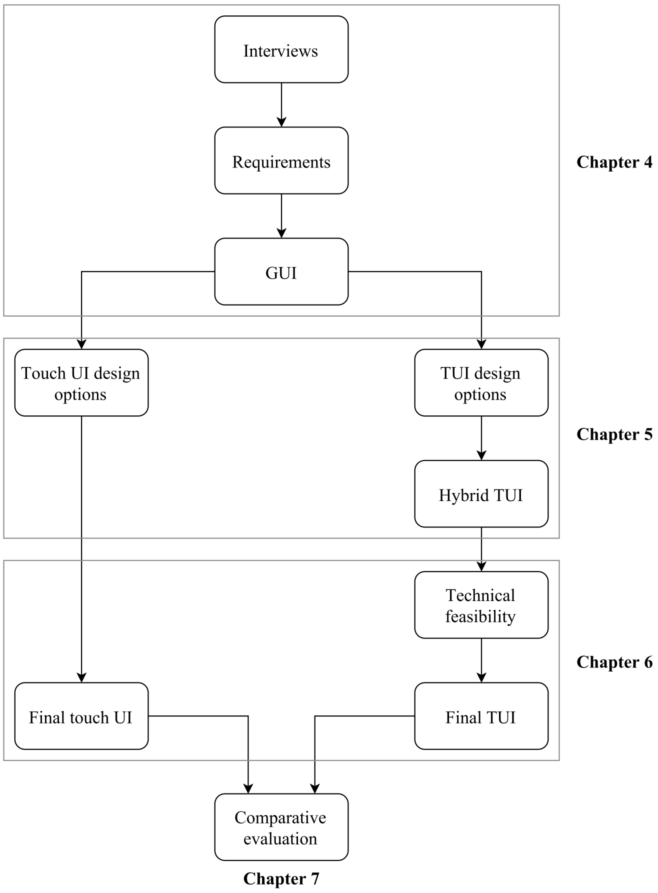


Figure 1.1: Steps taken by this research to design, develop and evaluate an infovis TUI.

and addresses some terms that help clarify quantitative genetics concepts. Then, quantitative trait loci (QTL), an analytical process for identifying regions in the deoxyribonucleic acid (DNA) that affect traits, is described. The thesis pays particular attention to eQTL, as the case study of choice, and introduces tools utilised for its mapping and visualisation. The chapter concludes with examples of graphical tools used for the identification of eQTL using non-traditional visualisations.

The main contributions of the research are described in **Chapters 4-7** (see Figure 1.1), followed by conclusions and future work (**Chapter 8**).

Chapter 4 This chapter investigates how quantitative genetics analysts interactively explore QTL and eQTL results. It then discusses the implications from semi-structured interviews and identifies high-level tasks and themes for quantitative genetics in general and for eQTL. A GUI is developed to explore novel functionalities and interactions for eQTL analysis. The chapter also investigates the effectiveness of a combination technique with touch and mouse inputs.

Chapter 5 In this chapter, the design options for developing a touch UI for interactive infovis is explored. The design considers gestures' flow and nature dimensions, as well as handedness. The chapter then poses arguments for the various techniques explored and their probable implementation in the baseline touch UI. Next, the design options for an exclusively tangible TUI are examined. The thesis then propounds arguments for the various levels of objects' specificity and their interactions. The chapter concludes by describing the design of a hybrid TUI that balances between touch and tangible interactions.

Chapter 6 This chapter identifies modalities for the interactions envisioned for the TUI in the previous chapter. It then investigates the technical feasibility of micro-controllers and sensors or computer vision for the detection of the identified modalities. The results show that computer vision is more effective for sensing the modalities expected for the envisioned TUI. The thesis later presents the developed TUI for interactive infovis. Results from an informal evaluation are used to improve and develop the final TUI. Finally, the thesis presents the improved TUI (see Figure 1.2) and the baseline touch UI.

Chapter 7 This chapter investigates the performance of the TUI for interactive infovis in comparison with the touch UI. It hypothesises the promotion of bimanual interaction and epistemic actions in the TUI and therefore an improvement in performance. The experimental method is presented. The quantitative and qualitative evaluation results are described. The results show that the infovis tasks are completed faster when using the TUI. The results also show that more effective strategies to explore patterns are adopted for the TUI, and fewer unnecessary analyses are performed. The experiment's results nullify the bimanual interaction hypothesis and confirm the application of epistemic actions with the TUI. Qualitative results also suggest that the TUI is more useful, easier to learn and more satisfactory than the touch UI.

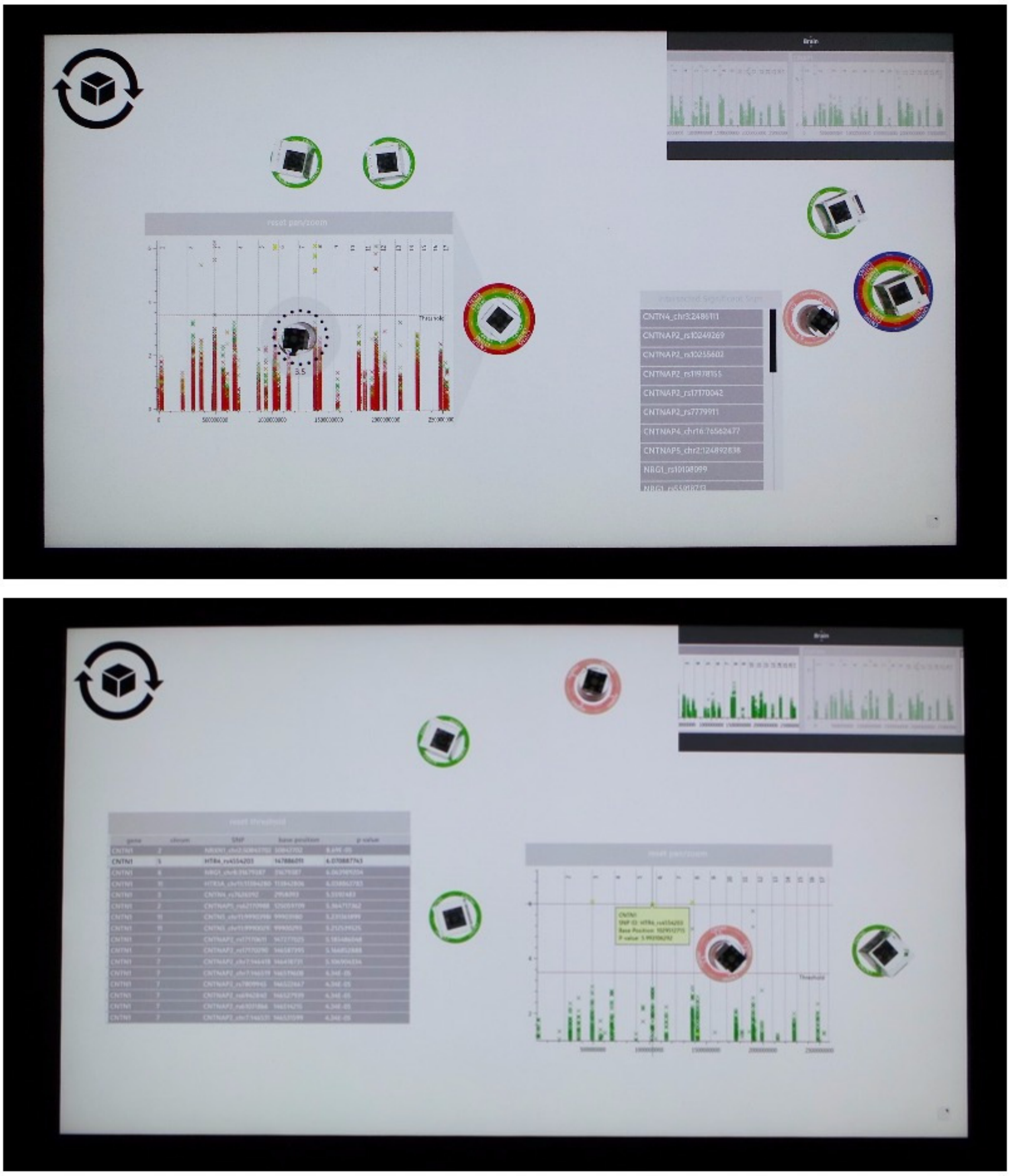


Figure 1.2: This research's infovis TUI.

Chapter 2

Background

2.1 Introduction

This chapter provides the background for the research described in this thesis and the context for the work. The chapter starts with a brief history of the utilisation of interactive surfaces and displays within the context of TUIs. TUIs and tangible interaction research is informed by theoretical foundations, which are discussed next in this chapter, that include affordance, bimanual interaction and epistemic actions [210]. A summary of various frameworks and taxonomies is also covered for the purpose of informing the design and development of the infovis TUI of this research. An overview of common technologies follows, which includes: computer vision, micro-controllers and radio frequency identification (RFID), along with prospective hardware and software toolkits.

TUIs are commonly claimed to enhance effectiveness and efficiency. Often, they are comparatively assessed against other interfaces, e.g. graphical and touch interfaces. Touch interfaces share similar characteristics with TUIs and have grown popular with recent users. They also provide a better baseline for comparison compared to GUIs. To that purpose, the chapter gives an overview of the various taxonomies and frameworks that characterise touch interaction to inform the design of this research's comparative baseline. Commonly, touch interfaces and TUI systems are combined to form hybrids, as described next.

Visualisation tools are integral to complex data analysis. Domain experts use highly interactive visualisation processes to explore data, investigate hypotheses and find patterns. Users perform literally hundreds of motor actions in the course of an analysis to access data, perform calculations, adjust parameters and manipulate windows [197]. The next section of this chapter introduces visualisation as an umbrella term and makes distinctions based on data type. The section frames infovis tasks to be implemented in a TUI or touch UI by summarising common frameworks and taxonomies and introducing the concept of direct manipulation. It finally surveys infovis TUIs to highlight diversity as well as common and recurring properties.

The final TUI and touch UI are assessed for performance differences and user preferences. A survey of common HCI evaluation methodologies is provided in this chapter. The thesis also describes common approaches for evaluation in the contextual domain of infovis. In particular, within the context of seven scenarios that span all stages of a visualisation tool's lifecycle [139]. The infovis TUIs surveyed in the previous section are further addressed here for the purpose of identifying limitations and common approaches for evaluation.

2.2 Tangible User Interfaces

Graphical interfaces have been the dominant interface type since the 1970s. With a GUI, a user typically sits at a desktop and interacts with the WIMP style interface using a mouse and a keyboard. It was not until the 1990s that Mark Weiser and colleagues from Xerox's PARC embarked on research that integrated computers seamlessly into the environment, now known as ubiquitous computing [257]. The work at PARC experimented with different sizes of tabs, pads and boards to suit particular tasks, resulting in ParcTabs, ParcPads and LiveBoards. It was with this vision that researchers were encouraged to explore novel post-WIMP interfaces that integrate the physical and digital worlds.

A TUI is defined as, '[...] an emerging post-WIMP interface type that is concerned with providing tangible representations to digital information and controls, allowing users to quite literally grasp data with their hands.' [210]. Ishii and Ullmer described the key idea of TUIs '[...] is to bridge the gap between cyberspace and the physical environment by making digital information (bits) tangible.' [105]. In this case bits are made more accessible using interactive surface, coupling of bits and atoms, and ambient media. Tangible interaction is used by Hornecker and Burr [100] as an umbrella terms for a wide range of interfaces that utilities embodied interaction, tangible manipulation and physical representation. TUIs, thus, fall under this umbrella. These varying definition are concerned with enhancing interactions using tangible objects at varying degrees of physicality and attention.

The following section provides an overview of the origins of the TUI concept, and a brief review of the most common TUI type (interactive surfaces [246]). This is followed by a description of theoretical foundations that inform TUI research. The next section summarises a selection of classifying frameworks that can inform the design and development of TUIs. TUIs are commonly developed using various technologies; the following section provides a summary of these technologies and prospective hardware and software toolkits.

2.2.1 Origin

In 1995, the idea of using physical objects (termed bricks) to manipulate virtual objects on a horizontal display surface was proposed by Fitzmaurice et al. [64]. The research introduced a new paradigm (known as graspable UI), where electronic content is physically grasped and directly manipulated. A graspable object was thus composed of one or more physical objects and a virtual object. Acting as specialised input devices, the bricks were tracked by the host computer while being physically handled by the system users. The physicality of the handle offered a variety of intuitive interaction techniques, such as bimanual and parallel interactions. An example of bimanual interaction would be using a brick to anchor the action of stretching a virtual square with another brick.

A distinction between space-multiplexed and time-multiplexed input was also introduced with graspable UI [64]. With a traditional computer set up, a mouse is typically time-multiplexed as it singularly controls all functions over time, one after the other. On the other hand, space-multiplexed input devices are dedicated to controlling one function each. Bricks display properties of both time- and space-multiplexed input. It is time-multiplexed as the mapping between a physical object and a virtual one was reconfigurable, and space-multiplexed as each virtual object was controlled by one or more physical objects, thus allowing parallel manipulation.

A number of studies [64] were carried out to explore the graspable UI concept and several benefits of the paradigm were discovered:

- Graspable UIs foster natural bimanual interactions.
- Graspable UIs advocate the use of a multiple input device, thus encouraging parallel interaction and expressive communication.
- Graspable UIs shift towards specialised context-sensitive devices that are efficiently tailored to a specific task.

A couple of years after the proposal of the graspable UI paradigm, the Tangible Bits [105] vision was introduced. This vision established a new HCI paradigm called TUI. TUIs augment the real world by coupling digital information to everyday physical objects and environments, rather than situating terminals into the physical environment (e.g. [257]). The work carried out for Tangible Bits [105] offered a broad collection of applications and interaction techniques that illustrated the various modes of coupling digital and physical spaces. This includes Tangible Geospace [242], which physically embodied landmarks from the MIT campus to allow users to navigate through 2D and 3D graphical maps of the campus, and ambientROOM [106], which utilised light, shadow, sound and airflow to communicate information to its users.

2.2.2 Interactive Surfaces

Interactive surfaces are commonly adopted in TUIs as a base for tangible interaction. Interactive surface TUIs are planar surfaces on which tangible objects are placed and manipulated [246]. At the same time, the interactive surface displays digital information to provide visual feedback. Interactive surfaces commonly mix technologies, touch and tangible, balancing the benefits and limitations of both. An interactive surface is typically horizontal (i.e. tabletop) or vertical (i.e. wall display). The potential advantages of using this form of interaction paradigm include the use of physical metaphors [1], space distributed input [63], external feedback and affordance [96], wider interaction space with continuous and direct interaction [88], support for learning [204] and collaboration [97].

One of the earliest examples of an interactive surface is the ActiveDesk, a projected digitised board that allowed for interaction other than touch [64]. This research pioneered the concept of graspable UI, now commonly known as TUI (see Section 2.2.1), where physical objects (bricks) are used to control electronic objects. metaDESK is an early platform used to explore the concept of TUI [242]. The interface enriched GUI controls with physical handles (phicons), as well as physical instruments such as active lens. Using projection, computer vision, magnetic-field position sensors and electrical contact sensors, the metaDESK was able to sense the phicons and physical instruments.

The I/O bulb is an evolutionary concept of the original light bulb, which projected information as a bulb would project light and collected video data of physical interactions [248]. The I/O bulb system was demonstrated as Illuminating light, a laser-based system that used holography and optics to physically model lasers, mirrors and lenses [249]. Urp was another demonstration of the I/O bulb for urban planning [250]. Utilising topological structures marked with coloured dots, an advanced vision technique was used to track tangible objects on a workbench.

Interactive horizontal surfaces have been used for musical performance with interfaces such as the Audiopad [185] and reacTable [114]. Audiopad is an electrical music composition system that combined dial-like controls with a multi-dimensional tracking interface [185]. The controls' movements were tabbed with RFID tags and detected via antennas. reacTable is a collaborative vision system by which physical musical controls were shared to synthesis music [114]. Fiducial markers were attached to the controls, which are tracked via the vision system, ReacTIVision [118]. Virtual information was displayed on the surface to monitor the state of the controls. Touch gestures were also utilised on the reacTable.

PlayAnywhere is a compact portable system that consisted of a projector, camera and infrared illuminants [262]. When placed on a flat surface, it projected digital information on a planar surface that could be manipulated via touch interaction and tangible objects. The infrared illuminates generated shadows detected by the camera to infer interactions on and above the surface. Luminos are physical building blocks, where each block contains fibre glass bundles that are stacked and clustered to form 3D structures [16]. Luminos extended the concept of fiducial makers to 3D in order to detect 3D constructions on a vertical diffuse illumination table. TZee is a passive 3D widget that utilises a diffused illuminations tabletops' capabilities to detect contact on three dimensions [261]. The widget allows for gestures on a tabletop to extend beyond the x- and y-axes and support z-axis transformations. This broadens the gesture space for touch and tangible objects and enhances interaction.

Sensetable is a system that used intelligent objects that were detected electromagnetically [184]. Information was projected on the objects' surfaces to allow for the clear identification of their functionalities. The system also utilised vertical displays that provided additional information relevant to the physical interaction. The sensetable application was demonstrated for teaching chemistry and system dynamics simulation [184]. A vertical simile of the sensetable is the senseboard [108]. A senseboard was used with data and command pucks to manipulate discrete pieces of abstract information. The system combined projection with RFID to identify and sense pucks' movements. Another vertical wall-sized display, HoloWall [160], used an infrared camera to detect touch and objects. Considering its early development, it was still able to detect multi-touch, hand and objects simultaneously.

Commercial interactive surfaces that support the detection of tagged objects have recently become available. Microsoft released MS Surface (later renamed Microsoft PixelSense), a tabletop technology with a multi-touch surface that senses interactions made by bare hands and/or objects. A second generation of Microsoft PixelSense was later released in 2011 with the Samsung SUR40. MultiTaction Cell Display is a multi-touch tabletop technology that is able to detect full hand interaction as well as distinct contact points. The display is also able to detect objects tagged with 2D bar codes on and above the surface.

2.2.3 Theoretical Foundations

TUIs are inspired by conceptual foundations that include affordance, bimanual interaction and epistemic actions [210]. The following sections analyse conceptual foundations that underpin the theorised advantages of tangible interaction and informs the design of TUIs. Experiments that evaluated performance to validate the conceptual foundations are also included when available.

2.2.3.1 Affordance

The term affordance was first coined by Gibson in his article, Theory of Affordance, in which he described the affordance of an environment as, ‘what it offers the animal, what it provides or furnishes, either good or ill’ [73]. The concept of affordance in the context of HCI was later introduced by Norman with an adjusted meaning to include perceived properties along with actual properties, ‘the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used’ [178].

In the case of TUIs, an object’s physical affordance is enabled by the physical nature of the object. Thus encouraging investigations into the object’s real and perceived affordance and how it can inform its choice, since the weight, shape and texture of an object affect how users may handle it. For example, Norman described a study in which three different puzzles were compared with one another by simply changing the objects used while the rules of the game remained unchanged [177]. The results of the study showed how the affordance (and in tandem the constraints enforced by the physical design of an object) affected completion time of the puzzle.

Similarly, a study was carried out to investigate the difference between children and adults in the interpretation of abstract and realistic objects used in a tabletop game [93]. The groups of game objects were categorised based on the amount of visual elements (relating to the game character) that the objects contained. The categories ranged from realistic, where a clear visual link existed between the object and the character, and abstract. The study results showed that all users, children and adults, better understood the function of a game object from objects with a high level of abstractness. This shows that only a few distinguishing elements of an object are required to be changed for function to be interpreted from an object.

2.2.3.2 Bimanual Interaction

People in their everyday interactions with the physical world utilise both hands to carry out tasks, such as changing gear while steering a car. However, this is not fully reflected in their interactions with a traditional computer setup, where a mouse is used by a single hand to carry out many tasks while the keyboard is used bimanually. In the case of TUIs, the use of multiple physical objects encourages the natural adoption of bimanual interaction to carry out tasks and thus better reflect everyday physical interactions (e.g. [64]).

Bimanual interactions are classified into two classes: bimanual symmetric interaction techniques, in which both hands work together with equal levels of importance at the same time (such as folding a sheet or skipping with a rope), and bimanual asymmetric, in which, within the non-dominant hand's frame of reference, the dominant hand can perform frequent actions as the non-dominant hand performs infrequent actions, such as writing with the dominant hand while the non-dominant hand manipulates the paper.

Guiard's kinematic chain model highlights the way labour is divided between the two hands [81]. Guiard states that there are three ways of assembling the two motors (i.e. hands): orthogonally, in parallel and serially. An orthogonal assembly involves the two motors governing two separate motions that are orthogonal to one another. With a parallel assembly, the two motors are in synergy, controlling the same motor proportion. Two motors that are assembled in a series work on the same motion, whereby one motor consumes the product of the other motor. He further explains that a serial assembly is what human hands mostly exhibit and thus model bimanual interaction asymmetrically. The assembly suggests that within the non-dominant hand's frame of reference, the dominant hand can perform frequent actions as the non-dominant hand performs infrequent actions. The model has since guided the investigation and evaluation of many bimanual interaction techniques (e.g. [19, 181]).

Two studies were carried out investigating the symmetric bimanual interaction class for improving performance by splitting subtasks and compound tasks between two hands [28]. The first experiment evaluated bimanual usage for a selection/positioning task using a bimanual interaction technique. With one hand the users were expected to position a graphical object, while the other hand scaled its size. Almost 41% of the users engaged in the parallel use of both hands, despite the training bias for serial usage. The results also showed the correlation between efficient user performance and the degree of parallelism utilised. The second experiment comparatively evaluated unimanual and bimanual interaction techniques for a selection/navigation task with novice and expert users. The results showed that the bimanual technique resulted in better performance by novices and experts, and the gap between the two users greatly reduced. However, only two users adopted the bimanual technique in parallel, i.e. symmetrically.

The conditions from the previous experiment [28] were later broadened to include two asymmetric bimanual interactions [116]: palette menu and Toolglass [19]. A palette menu was based on a painter's metaphor of holding a palette, where the painter holds the palette in one hand and uses the other hand to draw. Toolglass was similarly based on that metaphor but the menu was transparent with an integrated selection and initiation of action. The results of the study showed that it took less time to complete the task using Toolglass as it involved fewer motor operations and reduced cognitive load. That was not the case in the second asymmetric condition as it performed worse than others, and it was speculated that this was due to cognitive issues and not the lack of motor skills.

A few studies examined the potential benefits of symmetric bimanual interaction techniques, for example, a study observed the benefits of symmetrical techniques over asymmetric bimanual and unimanual techniques for an area sweeping task [142]. The increased benefit was attributed to mental load reduction and the increased degrees of freedom. For rectangle editing and navigation, symmetrical techniques performed better with a higher degree of parallelism than the asymmetrical techniques [33]. Another experiment explored the parallelism of a symmetric bimanual technique for standard target docking or selection tasks [11]. The results found that for these forms of interaction to be successful, an integrated task with a single focus is required. An empirical study compared bimanual and unimanual interaction techniques for a curve matching task [181]. The study showed that the task took less time to complete with the symmetrical bimanual condition, which was attributed, by the examiners, to the difficulty of the task.

The benefits of bimanual interaction, when designed appropriately, are not confined to TUIs because bimanualism has equally been promoted for multi-touch interaction with positive evaluated outcomes [51, 67, 268]. Nevertheless, the added third dimension in tangible object manipulation is viewed as a core advantage compared to other forms of interaction. The rest of this section summarises

Paper	Conditions	TUI category
Ullmer et al. [245]	TUI & GUI	Token+constraint
Terrenghi et al. [235]	TUI & touch UI	Interactive surface
Terrenghi et al. [234]	TUI & touch UI	Interactive surface
Tuddenham et al. [240]	TUI & touch UI	Interactive surface
Antle et al. [8]	TUI & touch UI	Interactive surface

Table 2.1: Experiments that investigated the effects of bimanual interaction in TUIs compared to other interfaces. TUI categories are based on Ullmer et al. [246].

studies that examine the performance of bimanual interaction in TUIs compared to a GUI and touch UIs (see Table 2.1). These studies also identify the types of bimanual interaction typically utilised for TUIs, since the previous results show varied utilisation of symmetric and asymmetrical bimanualism.

Tangible query interfaces (TQIs) use tangible objects to aggregate information, for example, by adjusting database parameters [245]. The objects (tokens or sliders on bars) were manipulated to adjust and visualise data sets. A preliminary user study compared one of the tangible interfaces against a GUI query interface. Users took longer to complete the task with the TUI condition compared to the GUI, but not significantly. The authors suggested that this was due to an additional setup requirement of the TUI. Two-handed interaction was adopted by 80% of the users in the TUI condition, but there were no reports of its adoption in the GUI.

Comparative studies were conducted to investigate the effect of manipulating physical and graphical objects in the everyday tasks of forming a puzzle and sorting photos [235]. Conducted on an interactive surface, the graphical objects were manipulated via multi-touch while physical manipulations were carried out in the TUI. The physical interface was found to be significantly faster for sorting photos but slightly slower when forming a puzzle. The findings also showed that unimanual interaction was predominantly used in both interfaces. When bimanual interactions were used with the touch UI, they were mainly symmetrical in nature. The physical system promoted the use of asymmetric bimanual interactions, which suggested that the physicality of the system supports the natural asymmetrical allocation of the hands compared to the touch UI.

On an interactive surface, tangible interaction was compared against touch using tangible and graphical versions of a PhotoLens for browsing and organising photos in tandem with a stylus [234]. The study results highlighted the forms of interactions adopted for both interfaces, which involved bimanual and unimanual interactions. Bimanual interactions were only infrequently observed and when they occurred they were largely noted for the touch version. This suggests that the permanence of the object made it easier for the user to leave it where it was once placed on the interactive surface.

Later studies compared tangible against touch interaction and a mouse and puck condition for a manipulation and acquisition task in which users were evaluated on performance and preference rankings [240]. Both experiments found better performances for the tangible condition and in both experiments the tangible input was preferred over the other two conditions. Performance analysis was extended to consider bimanualism and no effect was found for the physicality of the object on the adoption of unimanual or bimanual techniques. The experiments were video recorded and the analysis identified asymmetric bimanual interactions, and also what the authors referred to as concurrent unimanualism, whereby both hands were interacting concurrently but each on an independent physical control.

Another experiment compared touch and tangible interaction for a spatial problem-solving task, and collected quantitative and qualitative results [8]. As in other studies, the TUI condition performed better than the touch UI for the jigsaw puzzle task. Bimanual interactions were witnessed for both the touch and tangible conditions. For touch interaction, unimanual interactions were mostly used, but when bimanual interactions were utilised they were mostly symmetric in nature. Tangible bimanual interactions varied between symmetric, asymmetric and concurrent unimanual, and enabled effective epistemic strategies.

2.2.3.3 Epistemic Actions

Epistemic actions [127] (subsequently renamed complementary actions [130], but due to the popular use of the earlier term it will be used for the rest of this thesis) are activities external to the mind, that recruit environmental elements (e.g. hands or objects) to reduce cognitive load by adapting the world to our perceptual capacity. The relationship between an object's affordance and external representation on the adoption of epistemic actions has been highlighted in various researches (e.g. [125, 183, 231]). The external representation of an object helps users to actively employ that representation to their physical and cognitive advantage. By affording actions, reducing and limiting the complexity of interactions, epistemic actions are made easier and more intuitive [210].

The external activities of epistemic actions ideally provide information faster than an internal cognitive process would. Priming memory involves the perceptual identification of external entities by activating particular representations subconsciously before carrying out a pragmatic action, an action performed to bring a person closer to their goal. The role of epistemic actions in priming memory may raise the possibility of cueing information retrieval via external means. This in turn reduces the need for internal processing and results in improved performance [152, 154].

An example of a pragmatic action is moving a chess piece to establish check mate, whereby tentatively moving a piece to uncover further information (e.g. the consequence of moving a certain piece) is considered epistemic. Epistemic actions can improve cognition in the following ways [130]:

- Reduce space complexity by reducing the memory involved in mental computations.
- Reduce time complexity by reducing the number of steps in mental computations.
- Reduce unreliability by reducing the probability of error in mental computations.

The plausible benefits of epistemic actions were explored in a Tetris task using an empirical hazard function estimate to counter the cost of an epistemic action against the possible benefits [155]. The results showed that rotation previews of a zoid (a Tetris piece) reduced the response time needed to decide whether the zoid fitted a configuration, and this benefit outweighed the cost of the external action. The adoption of epistemic actions by Tetris players was claimed to increase

Paper	Conditions	TUI category
Patten et al. [183]	TUI & GUI	Token+constraint
Maher et al. [157]	TUI & GUI	Interactive surface
Antle et al. [7]	TUI, GUI & physical artefacts	Interactive Surface
Antle et al. [8]	TUI & touch UI	Interactive surface
Esteves et al. [58]	TUI & touch UI	Token+constraint

Table 2.2: Experiments that investigated the effects of epistemic actions in TUIs compared to other interfaces. TUI categories are based on Ullmer et al. [246].

with expertise [153]; however, this was not replicated for a later experiment [44]. Nevertheless, the cost of performing epistemic actions was investigated for skilled Tetris players and the results showed that the benefits outweighed the extra time needed to achieve an epistemic action [156].

The use of epistemic actions was found to guide thinking in various disciplines by organising graphical [171] or physical elements [82, 122] to externalise internal processes. The spatial arrangement of external elements, graphical or physical, via epistemic actions was one way to externalise thinking [43, 128, 171]. Another way involved using one's hands to gesture [4, 77, 129]. The use of external actions also led to the development of new perspectives and practices in education, e.g. playful learning [43, 54, 252].

In the context of TUIs, the relevance of epistemic actions lie in the physicality of the objects and their potential use as thinking aids. Even though the benefits of epistemic actions are not restricted to a tangible or physical system, the added physicality promotes the use of epistemic actions. The rest of this section summarises experiments that explored the adoption of epistemic actions in TUIs against GUIs and touch UIs (see Table 2.2).

A TUI was compared against a GUI in a location recall task of news articles and their relationships [183]. It was reported that in the TUI condition, participants performed better in the recall task while also performing more epistemic actions. In another GUI-TUI comparison for a design task [157], participants were noted to move objects in a trial-and-error fashion with the TUI more often than in the GUI condition. The TUI condition also sparked more discussions and speculations while gesturing and spatially constructing relationships. The experiment's results suggested that tangible systems were more effective for spatial designs as it was easier and more encouraging to form spatial relations between the design artefacts and the space.

Another experiment compared a TUI and a touch UI for a spatial problem-solving task (a jigsaw puzzle) from which quantitative and qualitative data were collected, as well as video recording [8]. The quantitative results showed a performance advantage for the TUI compared to touch. Qualitative observations suggested that the TUI encouraged epistemic actions that progressed towards mental problem solving, whereby touch prompted a trial-and-error approach. This was in line with an earlier study comparing TUI, GUI and physical artefacts, which found that the TUI enabled epistemic action early in the task [7].

In a problem-solving task that involved a four-in-a-row game, a TUI version was compared against mouse and touch versions for performance, mental projection, epistemic actions and workload [58]. The results showed that the TUI outperformed the mouse condition but not the touch. No significant difference in workload was observed between the three conditions. Epistemic actions, such as hovering or pointing with a finger, were noted for all users across all conditions. This casts doubt on the value of physicality in its support for epistemic actions.

2.2.3.4 Summary

Theoretical foundations inspire TUI research and inform the design of TUIs. Three conceptual foundations: affordance, bimanual interaction and epistemic actions were described and analysed generally and in context of UIs. To assert the benefits of these foundations for TUIs, comparative experiments were surveyed for bimanual interaction and epistemic actions. Various types of bimanual interactions were observed with TUIs [8, 234, 235, 240, 245] and touch UIs [8, 234, 235]. However, the general promotion of bimanual interaction in either interface was infrequent and put in question [234, 235, 240]. Epistemic actions were encouraged more with TUIs compared to other interfaces [7, 8, 157, 183]. These foundations guided the design of this research's infovis TUI so that it may encourage bimanual interaction and epistemic actions, which were later assessed (see Chapter 7).

Careful consideration was also taken to better afford meaning to the tangible objects utilised with the system.

2.2.4 Framing and Classifying TUIs

As TUI research matures into a discipline, various frameworks are developed for different stages of the TUI development lifecycle: abstraction, design and build [161]. Frameworks for abstraction provide classifications and taxonomies to map a system across different TUIs. Design frameworks provide tools that aid the design of a TUI by questioning or challenging the process. Frameworks for building provide steps and guidelines for developing TUIs. This section reviews a selection of the frameworks.

2.2.4.1 Frameworks for Abstraction

The graspable UI concept offered five core defining properties of graspable interfaces [63]. The primary property is the space-multiplexing of inputs and outputs, in which each function is controlled by a dedicated device with its own space and time. The four properties enabled by the space-multiplexing of inputs are:

- Concurrent use of multiple devices, which calls for inter- and intra-device concurrency, e.g. manipulating two or more devices to achieve a task.
- Specificity of input devices for the purpose of efficiency over generality; strong-specific devices increase efficiency due to their physical affordance.
- Spatial awareness of a device's surroundings; a device's awareness of other surrounding devices and its ability to communicate.
- Spatial configurability of devices based on context; a device's awareness of the surrounding environment that give meaning and purpose to its task.

The first steps undertaken to recognise TUI research as a concrete research field highlight the desired characteristics of such interfaces to develop the model-control-representation (physical and digital) (MCRpd), as an interaction model for systems with a TUI [244]. In the work, TUIs manifest digital information physically using objects as control and representation and thus eliminate the distinction stipulated by the GUI MVC (model-view-controller) model, on which

MCRpd is based. The MCRpd model highlights the key characteristics of TUIs as follows:

- Computational coupling of physical representation and digital information.
- Embodiment of interactive control by physical representation.
- The perceptual coupling of physical and mediated digital representations.
- Embodiment of fundamental features of the system's digital state by the physical state of the object.

The term MCRpd is later revised to model-control-representation (intangible and tangible) (MCRit) for clarity [247].

Approaches to TUI design are classified into three high-level groups [246]. Interactive surfaces allow the manipulation of the embodied physical objects on a flat augmented surface (see Section 2.2.2). This approach was popularly applied with applications such as Urp [250]. Another approach, constructive assemblies, was inspired by building blocks whereby a TUI is constructed by interconnecting modular blocks to model physical systems. A constructive assembly toy with a kinetic memory is an example of that approach [192]. The third approach, token+constraint, combines two kinds of physical object that can either be a token and/or a constraint, where tokens represent digital information and constraints provide structure and guide the interactions with those tokens (e.g. [245]).

The meaning of tangible objects used in an interactive surface are classified on a continuum [250] (see Figure 2.1). Noun objects, lying at the centre of the axes, are physical representations of their digital counterpart. As the classification moves to the right of the continuum, objects become more generic and abstract. Verb objects are manipulated to alter the digital representation in a way that is not related to their physical representation. Further along the continuum, reconfigurable tool objects are completely abstracted from the physical presentation. To the left of the centre of the axis, objects are stripped of what can be done with them. Attribute objects only consider one single attribute of an object when manipulating digital representations. Further stripping the object results in a pure object whereby its existence is the only representation required.

Frameworks have also been created for subsets of TUIs, whereby physical objects are associated with digital information that exists outside of the objects [98].

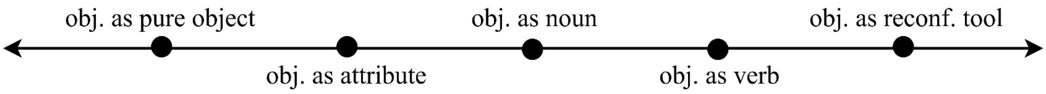


Figure 2.1: Tangible object's meaning continuum [248].

The framework introduces a schema with three types of linking object: containers, tokens and tools. Container objects are generic and used for the manipulation of digital information, e.g. to access or distribute digital content. While container objects are generic, token objects physically embody the information they represent. Tools, unlike containers and tokens, represent computational functions. Nevertheless, lines between these distinctions may sometimes be blurred. For example, a container object may also be used as a tool to control its content.

A later framework categorised TUIs within a 2D space of embodiment and metaphor [61]. Embodiment considers how closely tied the input is to the results of an interaction and is characterised in order of closeness as: full, nearby, environmental and distant. Metaphors are categorised in four levels: none, noun or verb, noun and verb, and full (similar to [250]). A TUI can adopt various levels of metaphor for interaction. This framework can be used to categorise TUIs as well as lending itself to design principles that guide future developments.

2.2.4.2 Frameworks for Design

The TAC paradigm [212], inspired by the token+constraint approach of TUI design [246], provides an understanding of the structure and functionalities of TUIs. The paradigm identifies five components to describe a TUI: pyfo, token, constraint, variable and TAC. A pyfo is a physical object that takes part in a TUI and could either be a token or a constraint. A token is a graspable pyfo that is coupled with digital information or a computational function and whose behaviour is limited by another pyfo (known as a constraint). The term variable is used to describe digital information or a computational function in an application. A TAC defines the relationship between a token and its variable with one or more constraints. The TAC paradigm is applied by defining relationships between a token and its variables with one or more constraints.

A framework structured around four interrelated themes addresses the broader design space of tangible interaction and provides support for social interaction and collaboration [100]. The framework provides perspectives to guide the conceptual design and assessment of tangible interactions, which encompasses TUIs. Tangible manipulation, the first theme, involves the direct manipulation of the material objects that are coupled with digital information. Spatial interaction focuses on the occupancy of tangible interfaces of real space. Facilitation methods that support and guide social configurations are the concepts highlighted in the third theme - embodied facilitation. Finally, express representation refers to the expressiveness of objects as representations of digital information.

One model of the tangible interaction process aims to link three levels of reality (cognitive, physical and virtual) to allow for the formation of critical transitions in the process of interaction [95]. With the actor and tangible interaction system as prerequisites to the process, interaction cycles are formed following a number of steps: perception of the object and its context, the planning and interpretation of an action, performing an action on the object, and the effect of that action and the system's feedback in the previous steps. Effects include physical, virtual, status and remote. The model also considers multi-user scenarios and introduces external actions and effects that manifest separate interaction cycles.

Material probe is a design approach that explores how people perceive the materials used with tangible objects by comparing and contrasting the quality [115]. This approach aims to aid designers in considering the shape, colour, texture, and historical/cultural reference when developing tangible objects. The design implications suggested by material probe are namely material simulation, expression and exploration. Material simulation is the emulation of sensory feelings when performing an action with an object. The object's expressiveness of affordance or social presence is implied with the material's expression. Material exploration recommends the adoption of material investigation early in the design phase to create new forms of digital content.

Another compare and contrast approach, conceptual metaphor theory, considers the relations between physical objects and their abstract representations [150]. Their experimentation validated twenty conceptual metaphors, e.g. big is important, that provide guidelines to inform the design of a physical object in a tangible system. Similarly, the relations between the physical object and its discrete or continuous digital function is explored [188]. The mapping considered provides guidelines for the linking of physical objects to digital content, which includes: mapping discrete functions to a flat surface, continuous values to a convex surface and indeterminacy of a digital function's value to the edge of a surface.

2.2.4.3 Frameworks for Build

Frameworks for building TUIs consider prototyping hardware/software, as well as toolkits that support the development of sensing TUIs. Toolkits provides the means to rapidly explore design options by physically informing the feasibility of these ideas. Section 2.2.5 describes various toolkits that support the development of TUIs along with programming tools to interface the physical devices with digital information.

Three types of rapid prototyping techniques considered for two example TUIs explore form and interaction using off-the-shelf digital devices [186]. The three techniques include embedding an off-the-shelf device into a new form (embedding), partial use of its content in a new device (cracking it open), and combining multiple devices to create a complex prototype (collating). The process describes a general framework that guides rapid prototyping to improve the communication of ideas and to explore interactions.

The framework's [186] process is composed of five steps: concepts to explore, selection of existing technologies, deciding on a form factor, evaluation and reflection. The developer first starts with a concept and identifies its interesting aspect. After a concept has been identified, existing technology is surveyed and compared. A form factor is then decided on, whereby the concept is refined and reformed to take technology into account. Once a prototype is developed from the form factor, it is used by the developers to evaluate technical feasibility, user experience and performance for the purpose of reflection.

2.2.4.4 Summary

Various frameworks were developed for different stages of TUI development (abstraction, design and build) [161]. In this research, frameworks were applied to provide a conceptual structure to think through the TUI and its components at various stages of its development. Frameworks for abstraction were used in this research to classify tangible objects [250] and the TUI system as a whole [246] (see Section 5.3). Design frameworks aided the TUI design process by questioning and challenging it. The TAC paradigm [212] was utilised for the design of the TUI as it classified the system's components and objects and defined the relationships between them (see Section 5.3). A design framework was also used to map the objects' abstract representation to infovis digital functions [188] (see Section 6.4.1). Frameworks for build provided guidelines for developing the TUI and toolkit support; a number of electronic toolkits were acquired and assessed for their suitability at sensing the TUI's modalities informed by the design (see Sections 6.2 and 6.3).

2.2.5 Implementation Technologies

Common technologies predominantly used to implement TUIs include: RFID, computer vision and micro-controllers [210]. Often, two or more of these technologies are used to develop a TUI (e.g. [9, 108, 253]). This section provides a summary of these technologies and an overview of prospective hardware and software toolkits for TUI implementation.

2.2.5.1 Radio Frequency Identification

RFID is a radio-based wireless technology that uses electromagnetic signals to determine the presence and identify of a tagged object within the range of a tag reader. RFID tags are comprised of an electronic circuit for storing data, and an antenna for receiving and transmitting a signal; once a reader broadcasts radio waves, the tags communicate and identify themselves accordingly and only when in range. Tags can either be passive or active. Passive tags have no built-in power supply and rely on the energy of the radio waves transmitted by the reader.

Paper	Error	
	Orientation (degrees)	Position error (cm)
Hahnl et al. [85]		100-140
Jin et al. [112]		72
Zhang et al. [270]		100
Vorst et al. [255]		20-26
Joho et al. [113]		35.5
Hekimian-Williams et al. [94]		1
Ting et al. [237]		7%*
Shirehjini et al. [170]	1.9	6.5
Dao et al. [42]		32.3
Aguilar-Garcia et al. [2]		1000

Table 2.3: Experiments that evaluated the performance of RFID at detecting position and/or orientation ($\star$ error rate was only provided as a percentage).

Active tags house batteries to generate radio waves that can broadcast even in the absence of a reader [76].

Various techniques are applied for the detection of a tagged object's position and/or orientation using active or passive tags with varying results (see Table 2.3). Tag range estimation techniques detect the position of an RFID tag by evaluating readings from one or more RFID reader antennas [271]. Range estimation can be derived from phase and time measurements or through received signal strengths (RSS) [2, 42, 94, 113, 237]. The accuracy of such techniques is dependent on the range estimation algorithm used. Phase measurements estimate distance by using a signal's path-of-arrival delay sensed with transmitters at static locations [94]. Time estimation methods consider the distance between multiple receivers and the tagged object and either estimate the distance based on the propagation time or by using the difference in time from multiple receivers. RSS-based approaches estimate distance by localising the emitted signal strength between the tag and at least three readers and is one of the most commonly adopted approaches [2, 42, 113, 237].

Direction-of-arrival (DOA) estimation techniques observe DOA information from multiple RFID reader antennas to localise RFID tags. This is typically achieved with phased arrays, directional or smart antennas. The arrays processing approach uses the phase measurements of a tagged object from multiple antennas to determine distance [270]. The use of directional or smart antennas is a less accurate alternative for the detection of distance compared to phased arrays. Another technique utilises reference tags and constructs comparisons of detection rate or distribution probability of unknown RFID tags against the known reference tags [112, 170, 255].

Table 2.3 shows the possibility of locating tagged objects in an indoor environment, however the error rate is too large for a TUI given that other implementation technologies can provide better performance. RFID technology has been used in TUIs predominantly to detect the presence and determine the identify of tagged objects. Their low cost and small size makes RFID an attractive prototyping tool. Example applications include senseboard [108], where tagged pucks are placed on reader slots to be identified, and a TQI [245], where constrained tagged tokens are used to control database parameters.

2.2.5.2 Computer Vision

Computer vision, as the name implies, equips computers with vision abilities to reconstruct, interpret and understand a scene. The technology is becoming increasingly popular for perceiving people, objects and scenes in robotics, transportation and medicine. One approach in computer vision acquires digital images via cameras to understand the world. It then applies image processing algorithms to detect distinct objects in the digital images [68]. Marker-based vision (MBV) systems involve another approach that identifies and locates fiducial markers in space [193]. Objects are tagged with barcode-like tags to uniquely identify their position, orientation, shape, etc. The MBV approach is the prevalent method used for developing TUIs due to its robustness, improved performance, reduced cost and increased reliability [210].

A typical computer vision TUI employs a projector and a camera, which is mounted above or below a projection surface. Many displays described in Section

2.2.2 employ the computer vision method to detect objects on planar surfaces. Urp uses small coloured dots to uniquely identify topological objects representing buildings and roads as well as tools in an urban design and planning system [250]. In a similar fashion, PlayAnywhere is able to detect an object's position and orientation by uniquely identifying a 12-bit code and an object's strong edges [262]. For the detection of stacks and clusters in a computer vision system, Lumino uses 3D markers constructed from glass fibre bundles [16].

Several toolkits for computer vision are readily available for prototyping. The ARToolKit [123] is a collection of C and C++ software libraries used for developing augmented reality (AR) applications (e.g. [20, 71, 209]). The toolkit tracks the position and orientation of square fiducial markers using video tracking capabilities. Papier-Mache is a Java-based software toolkit that supports the development of tangible systems by detecting objects tagged with fiducial markers, barcodes or RFID [132]. Another toolkit is reacTIVision, a vision-based framework, that is written in C++ and tracks tagged tangibles on a table [118]. It is the primary sensor component for reacTable [114] and several other tangible systems (e.g. [159, 223]). ToyVision [158] and TULIP [238] are software frameworks that extend existing vision frameworks; ToyVision aims to facilitate the implementation of tangible systems for designers and developers [158], and TULIP employs the MCRit [247] abstracting framework to ease rapid prototyping of TUIs [238].

A number of commercially available vision-based tabletops are capable of detecting tagged objects on their surfaces. ReacTable is a collaborative system for vision used to synthesise music using tagged music controllers and multi-touch gestures [114]. Microsoft PixelSense, a vision-based multi-touch display, detects finger and objects contacts as well as tracking 2D visual markers (known as byte tags). A later instalment of Microsoft PixelSense is the Samsung SUR40 display. MultiTaction cell display is able to identify touch/hand input and track tagged objects. The latter two displays can be set up as a wall or as a table, inclinable to any position.

2.2.5.3 Micro-controllers, Sensors and Actuators

Micro-controllers are small computers that are embedded onto physical objects to act as a gateway between the physical world and the digital world. Information is received by the micro-controller from the physical world through sensors. Sensors capture a wide range of physical properties, such as motion, acceleration, distance, temperature, and so on. Actuators in turn can affect the physical world by producing motion, sound, light or haptic feedback [210].

The adoption of micro-controllers facilitates the development of tangibles, i.e. active or intelligent, that input information wirelessly to a display or act independently. Easigami is a tangible system for constructing polyhedral 3D objects using active polygonal tangibles [101]. Each polygonal and connecting hinge is equipped with a micro-controller and LED pins to aid the construction of a polyhedral. One of the TQI system's expresses database queries using parameter bars and a graphical monitor [245]. The bars are active tangibles made up of a micro-controller, LCD display and dual slider sensors to manipulate the bounds of a query variable.

Several electronic toolkits for constructing active tangibles are commercially available and are straightforward to use. Arduino is a physical computing platform that offers an array of I/O boards [21]. Various sensors and actuators are attached to an Arduino board to construct sensing/actuating physical objects, which are programmable via the Arduino API. Phidgets [80] and Calder [141] are used in a similar fashion as the Arduino [21]. Higher-level toolkits, such as LEGO Mindstorm [131] and littleBits [18] lessen the expertise needed to construct automated objects and thus facilitate faster prototyping. LEGO Mindstorm offers numerous devices that are connected and programmed through a prototype [131]. littleBits offers preassembled hardware components that are linked together to form more complex structures via magnets [18].

A number of software toolkits can also be used by designers and developers to construct tangible system prototypes rapidly. iStuff, while similar to low-level hardware toolkits such as Phidgets [80], adds a level of abstraction through PathPanel [12]. This intermediary software lets designers formulate high-level events that are dynamically mapped to I/O devices. D.tools facilitates the visual prototyping of physical systems using state diagrams [90]. Various I/O devices are represented graphically and arranged to construct physical devices. When connected to the physical devices, it is possible to programme hardware behaviour.

2.2.5.4 Summary

Three technologies are commonly adopted for the implementation of TUIs (RFID, computer vision and micro-controllers) [210]. In this research, implementation technologies are explored and evaluated to determine their feasibility at detecting a number of sensing modalities (see Table 6.1). The idea of using RFID technology was eliminated early in this research, due to its inaccuracy at detecting position and/or orientation (see Table 2.3). MBV was used to implement various TUIs (e.g. [114, 242, 250]) and commercial versions were commonly utilised (e.g. [55, 147, 204]). Micro-controllers, sensors and actuators link the physical and digital worlds and are typically used to implement entertainment and learning TUI systems (e.g. [101, 192]). Several electronic toolkits were considered for implementing this research's tangible objects, one of which was systematically evaluated for technical feasibility (see Section 6.2). A commercial MBV system (the Samsung SUR40) was similarly evaluated (see Section 6.3). Based on the evaluation results, the technology best suited for this research was used to implement the infovis TUI (see Section 6.4).

2.3 Touch User Interfaces

Multi-touch interfaces recognise two or more touches simultaneously, feasibly accommodating two or more people interacting with the touch system. This type of interaction is intuitive and is a natural form of contact and manipulation. The following section provides a brief history of touch interfaces. The next section classifies gestures that can inform the development of the baseline touch UI. A literature overview of touch performance compared to mouse input follows.

2.3.1 Brief History

Multi-touch technology dates back to 1982 with the flexible machine interface [163]. In 1984, at Bell Labs, the first multi-touch screen was developed [27]. The screen enabled the manipulation of graphical objects, which was possible with the transparent array of capacitors overlaid over a cathode ray tube. In the same year, a technological invention for touch sensing that used optical methods was patented [121]. The DigitalDesk, a computer vision touch UI, was later pioneered [259]. It was noted that the DigitalDesk was the first multi-touch interactive device that used some of today's familiar interactive gestures, such as pinch to shrink and spread to expand [201].

The 1990s and early 2000s saw an increase in multi-touch research carried out on handsets, tablets and tables. The flip board was introduced in 1992, and it combined a keyboard and a touch tablet with the aim of extending the capabilities of existing direct manipulation systems [25]. In the mid-1990s, a drafting table with a projected display was developed [26]. It was able to capture the position, pose and orientation of a user's hands. Bimanual interaction was enabled with the additional use of a stylus. In 2001, Mitsubishi unveiled DiamondTouch, which was able to distinguish between different collaborators' interactions [46]. Microsoft later introduced PlayAnywhere in 2005, an interactive tabletop, which uses a compact implementation of a computer vision system [262].

Multi-touch technology continued to evolve with little everyday use until recently. In 2007, Apple Inc. introduced the iPhone and iPod Touch, with a capacitive touchscreen for multi-touch sensing. It was then that the notion of multi-touch displays was popularised commercially. The introduction of the iPhone to the market and its widespread adoption led other chief handset manufacturers, such as Nokia and Sony Ericsson, to launch their own touchscreen handsets [201]. Microsoft later released MS Surface (later renamed Microsoft PixelSense), a tabletop technology with a multi-touch surface that senses interactions made by bare hands and/or objects. A second generation of Microsoft PixelSense was later released in 2011 with the Samsung SUR40.

2.3.2 Frameworks and Classifications

There has been a great number of studies that focus on multi-touch systems and interactions, as well as commercial activity e.g. touch phones and tablets. The potential advantages of gesture and touch input include their naturalness and support for space distributed interaction e.g. bimanual interaction. The rest of this section reviews a selection of touch and gesture classifications and taxonomies.

Gesture interaction techniques are categorised into five gesture styles [119]:

- Gesticulation gestures
- Semaphore gestures
- Manipulation gestures
- Deictic gestures
- Language gestures

Gesticulation is the natural non-learned form of gesturing that is commonly used in combination with speech to clarify meaning or description. Semaphores, in the other hand, are learned static or dynamic poses that are programmed and recognised by a system. Manipulative gestures are dynamic hand interactions that are tightly coupled with the manipulation of virtual objects. Gestures that involve pointing to identify virtual objects or establish their spatial location within a system are deictic. Finally, sign language gestures are grammatically and lexically mature and thus comparable to speech [119].

A taxonomy of surface gestures based on user behaviour is used to describe the gesture design space. Interactive surface gestures are classified along four dimensions: form, nature, binding and flow (see Table 2.4) [264]. Form dimensions are concerned with a single hand's static or dynamic pose in the context of either unimanual or bimanual gestures. The nature dimension categorises gestures into symbolic, physical, metaphorical and abstract based on their metaphorical nature to the gestural effect. The binding of gesture and location is either object-centric, world-dependent, world-independent or of mixed dependencies. This depends on the location of the effect within and outside of the system's context. A gesture's flow is considered discrete or continuous based on whether the effect is recognised then responded to or continually recognised.

Multi-touch gestures can be characterised based on degrees of freedom, spatial occurrence, semantics, trajectory complexity, number of users, number of fingers and timing [39] (see Table 2.5). Semantics generally map to nature from the taxonomy of surface gestures [264]. Trajectory complexity classifies gestures based on their shape and trajectory, e.g. an example of a closed trajectory would be tracing a path to form a circle. The timing property considers single (atomic) or a sequence of single gestures (sequential). These properties were mainly constructed

Dimension	Category	Example gesture
Form	Static pose	Press with hand
	Dynamic pose	Spread fingers
	Static pose and path	Drag hand
	Dynamic pose and path	Rotation by moving two fingers
	One-point touch	Press
	One-point path	Double tap
Nature	Symbolic	Tracing question mark for help
	Physical	Push away
	Metaphorical	Swiping to turn a page
	Abstract	Triple tap for action
Binding	Object-centric	Shrink gesture
	World-dependent	Dragging an object off-screen
	World-independent	Hold and tap to group
	Mixed dependencies	Asymmetric two-handed gesture
Flow	Discrete	Tap on menu item
	Continuous	Drag and drop

Table 2.4: Taxonomy of interactive surface gestures with examples [264].

to examine frameworks and techniques for gesture recognition, however they also prove useful in describing touch interaction techniques. A proposed classification to describe a touch gesture could include the number of hands used to perform an action.

2.3.3 Touch versus Mouse Input

There is a growing body of literature that compares various input devices for selection and/or dragging tasks. Table 2.6 summarises the major studies that compared computer mice and touchscreens for selecting and/or dragging a graphical item. The majority of the studies showed that the touchscreen (be it laptop sized or tabletops) was the fastest for either selecting a single item or selecting then dragging that item to a predetermined position. Target selection, menu selection and selection with typing tasks were used to comparatively assess a touchscreen,

Properties	Classes
Degrees of freedom	2D or 3D
Spatial	On the surface or above the surface
Semantic	Symbolic or direct manipulation
Trajectory complexity	Open, closed or crossing gestures
Number of users	Individual or collaborative
Number of fingers	Single-touch or multi-touch
Timing	Atomic or sequential

Table 2.5: Gesture classes by properties [39].

mouse, and keyboard in two experiments [120]. The results from both experiments showed that the touchscreen was significantly superior in speed to the mouse and keyboard. The participants also preferred the touchscreen or keyboard over the mouse for all tasks.

For an interactive encyclopaedia, a touchscreen, a mouse and two variants of key inputs were compared for target selection and path traversal [180]. Speed advantage was reported for the touchscreen over indirect input. This study also found the touchscreen to be the least accurate (difference not significant), but the preferred means of interaction among participants. Similarly, two studies compared a touchscreen and a mouse and found speed advantage for the touchscreen over the mouse for a command selection and a target selection task [47, 208]. A more recent study compared the mouse input and a touchscreen for a target selection task on a tabletop display [202]. The results mirrored those previously mentioned on smaller devices, where the touchscreen was found to be the fastest, albeit the least accurate. Unlike the study carried out in [208], the participants preferred the touchscreen over the mouse.

The performances of five input devices (which included a touchscreen and mouse) were comparatively assessed for a series of goal-directed tasks using a drawing programme [165]. The mouse significantly outperformed the touchscreen in speed and user preference. In contrast, in a drawing task, favourable results were reported for speed when using the touchscreen and better accuracy was obtained for the mouse [99]. The performance of older and younger adults was

Paper	Speed	Accuracy	Preference
Karat et al. [120]	Touchscreen		Touchscreen
Ostroff et al. [180]	Touchscreen		Touchscreen
Dillon et al. [47]	Touchscreen		
Sears et al. [208]	Touchscreen (16 px) Mouse (1 px)		Mouse
Meyer et al. [165]	Mouse		Mouse
Forlines et al. [67]	Touchscreen (selection) Mouse (dragging)	Mouse (selection)	Mouse
Sasangohar et al. [202]	Touchscreen	Mouse	
Hooten et al. [99]	Touchscreen	Mouse	
Cockburn et al. [41]	Touchscreen (selection) Mouse (dragging)	Mouse (selection)	
Findlater et al. [60]	Touchscreen	Touchscreen	

Table 2.6: Studies that compared touch and mouse input for the the selection and/or dragging of graphical items. The majority of studies compared other input devices against touch and mouse, but these are not reported in the table. Only significant results are included.

compared on a desktop and a touchscreen in a task that involved pointing and dragging [60]. The touchscreen outperformed the mouse in speed and accuracy for both age groups. Participants in both age groups also preferred using the touchscreen over the mouse. The contrasting results of these two studies with earlier research [165] is inarguably owing to technological improvements in the touch input.

The difference between the performances of the touchscreen and mouse were investigated using a tabletop display for a selection and docking task [67]. The task required the completion of both selection and dragging actions. The first experiment explored unimanual interaction, where the touchscreen outperformed the mouse in selection time, whereas the mouse outperformed the touchscreen in dragging and docking actions. The mouse was also significantly more accurate in selecting targets and was predominantly preferred by the participants. Another study compared the merits of a mouse, stylus and touch input for pointing activities, which included tapping and dragging [41]. The speed results mirrored those of Forlines et al. [67] with the touch input outperforming the mouse input for tapping actions and the opposite was the case for dragging actions. The touch input also proved to be the least accurate for dragging.

Most studies (see Table 2.6) that comparatively assessed the mouse and touch inputs for selecting and/or dragging a single item found favourable results for touchscreens, albeit with decreased accuracy. The selection of elements proved the fastest with touch input [47, 99, 120, 180, 202, 208] and was generally the favoured type of input [120, 180, 202]. For tasks involving combined selection and dragging, the touchscreens outperformed the mouse input in later studies [60, 99] but not in an earlier investigation [165]. Isolating the results of these two actions in other studies revealed that the touchscreen is the fastest for selection and the slowest of the two for dragging on a tabletop [67] and on a touch laptop [41].

2.3.4 Summary

Traditional workstation setups and GUIs are good benchmarks to comparatively evaluate TUIs (e.g. [157, 183, 245]). However, the focus has been shifting towards more natural input devices and technologies such as multi-touch (e.g. [149, 235, 240]), where the comparison gap is smaller and arguably more credible. In this research, a touch UI baseline is used to evaluate the infovis TUI (see Chapter 7). The design of the touch UI was supported by gesture frameworks and classifications to explore the gesture space and make informed design decisions (see Section 5.2). The classification of interactive surface gestures [264] was utilised to asses the nature and flow of gestures considered for the touch UI. Form was simplified to address handedness only, and the binding domain was not considered. Due to the abstract nature of this research's case study (see Chapter 3), symbolic and physical gestures were not utilised either. Studies that compared touch and mouse input found faster performances with touch for target selection [47, 99, 120, 180, 202, 208] but with decreased accuracy [41, 67, 99, 202]. Touch input was also found by some studies to outperform mouse input in selection tasks, but not dragging [41, 67]. The results of these studies were used to assess an eQTL task performance with the touch and mouse inputs to address the implications during the design of the touch UI baseline (see Section 4.4).

2.4 Hybrid User Interfaces

Multi-touch interaction displays are commonly implemented using a computer vision approach. These types of displays are capable of detecting touch as well as tagged objects that are placed on their surface. Both touch and tangible interaction promise direct, natural and easy to learn manipulations. Both also encourage bimanual interaction, space-multiplexing and parallel input. Table 2.7 summarises the similarities and differences between touch and tangible interactions.

Hybrid interfaces consisting of tangible and touch interaction are commonly seen in TUI literature. Interactive surfaces [246] are the typical type of TUI that integrate touch with tangible interaction. HoloWall [160], PlayAnywhere [262] and reacTable [114] are some of the interactive displays recalled in Section 2.2.2 that detect and utilise touch gestures. For instance, reacTable [114] adopts touch interactions to adjust the internal parameter of a music synthesiser or mute connections between different synthesisers.

Several other interfaces integrate touch and tangible interactions in various application domains. G-nome Surfer is a predominately touch interface used to navigate, access external resources and compare genome data [215]. It utilises tangible objects for two of its versions to compare genomes [211] and access glossary terms [214]. Facet-streams integrates touch and tangible interaction to

Property	Tangible	Touch
Input	Space-multiplexed	Space-multiplexed
Manipulation	Direct	Direct
Space	3D	2D
Feedback	Visual and tactile	Visual
Handedness	Bimanual/unimanual	Bimanual/unmanual

Table 2.7: The properties of touch and tangible interaction.

query a database [111]. To form a search criterion, a user selects a tangible facet and places it on the interactive surface to choose an option via touch. Another example, utilises tangible objects to represent people and objects in a police incident, where touch interactions are used to navigate the node-like visualisation and create connections between nodes [147].

VPlay and Family Archive are hybrid interactive surface that are implemented to develop guidelines for developing hybrid interfaces [126]. These guidelines address issues relating to balancing touch and tangible interactions on a hybrid interactive surface. While developing the two cases studies, two sets of decisions were outlined to develop the guidelines: (a) how to assign interface elements to digital or physical objects? (b) how digital elements can emulate interactions in the physical world?

Several implications and issues were raised durig the development of VPlay and Family Archive, which are summarised into the following [126]:

- Nature of the physical objects
- Eyes-free control
- Affordance of physical tools
- Affordance of digital tools
- Controlling a 3D world on a 2D surface
- Incorporating 3D manipulation
- The 3D world from a 2D techno-centric viewpoint
- Physical objects as containers
- Loss of physical objects
- Mode errors and physical feedback

These implications suggested that physical objects are more suitable for situations where eyes-free, precise control is required. They also suggested the utilisation of real-world affordance and incorporating it into the system's digital and physical objects.

Some of the limitations of TUIs include: clutter, loss of objects, fatigue, versatility and malleability [210]. A balanced hybrid interface would potentially alleviate these problems and serves to guide users' interactions. The balance between physical and digital representations is one of TUIs' greatest design challenges [126, 234, 244]. In this research, a TUI that attempted to balance between touch and tangible interaction for infovis tasks was envisioned (see Section 5.4). The design of the hybrid UI was informed by the implications derived from implementing VPlay and Family Archive [126].

2.5 Visualisation

Visualisation is the best means to explore and understand large data sets in a visual manner to find patterns, relationships and to verify and check data. Visualisations allow users to form and construct a mental image or vision of the graphical representation or to imagine or remember as if actually seeing [91]. Data visualisation is commonly used as an umbrella term that covers scivis and infovis.

Scivis is defined as a graphical approach that, '. . . allows researchers to observe the results of simulations using complex graphical representations.' [84]. This type of visualisation is mostly concerned with the graphical representation of volumetric spatial or spatiotemporal discretised data. It combines computer graphics, mathematical models and numerical methods to visualise scientific data. Infovis is defined as, 'The use of computer supported interactive, visual representations of abstract data to amplify cognition' [32]. Infovis represents abstract concepts that do not have physical aspects that lend themselves to the visualisation. Simple and effective interaction with infovis increases a user's understanding by utilising entrenched human experience. This was exemplified in the application of interaction to standard static graphical representation, e.g. pie charts and multi-line plots, which produced more efficient representations [49].

Infovis data types are categorised into seven classes by Shneiderman et al. [220]: linear, planar, volumetric, temporal, multi-dimensional, hierarchical and network. Linear data is one dimensional, typically textual and organised by a single feature in a list. Planar or map data types are two dimensional and include geospatial maps and floor plans. Volumetric data is generally skewed towards scivis as it represents real-world objects, such as molecules. Timelines are temporal data types that represent chronological lists of events. Bar charts, scatter plots, histograms, etc., are examples of multi-dimensional data types that are manipulated on the n$^{\text{th}}$ dimensional space. Tree structures and hierarchies connect a collection of data items with a single parent item. When data items are linked to an arbitrary number of items, then the data type is categorised as a network.

The thesis's case study is abstract in nature (see Chapter 3); therefore, the rest of this section only expands on infovis. Various infovis taxonomies and frameworks are considered in the next section for the purpose of framing the infovis tasks elicited from case study experts. Current infovis systems are largely WIMP interfaces, which provide direct interaction but comparatively lower than what is provided with touch and tangible interfaces. Therefore, the next section addresses direct interaction with infovis by providing principles and categorical phenomena. Compared to other application domains (e.g. in touch UI), infovis has had little research utilising TUIs. These studies are surveyed and categorised based on common features. Finally, the section is summarised and related to this thesis.

2.5.1 Infovis Taxonomies and Frameworks

The infovis pipeline is the process of converting information into visual form, whereby interactivity allow users to alter that pipeline at any of its stages to augment visual cognition and understanding [32]. Interaction in the context of infovis is concerned with reflecting change on the visual representation rather than entering or inputting data to the system [267]. While interactive infovis is used in many different application domains, there are similarities in the underlying tasks that users perform. This has allowed researchers to characterise the analysis process in many taxonomies and interaction frameworks, most notably in Shneiderman's visual information-seeking mantra, 'overview first, zoom and filter, then details-on-demand' [220], and the analytical tasks that users perform (e.g. [5, 22, 38, 92, 136, 145, 256, 267]).

Low-level steps that occur during infovis are derived with an affinity diagramming approach [5]. First, the work of infovis students is reviewed to gather a corpus of 196 analysis questions and tasks. An affinity diagramming approach is then used to group similar questions and refine those group iteratively. The taxonomy resulting from this is a set of ten primitive analysis tasks, partly in line

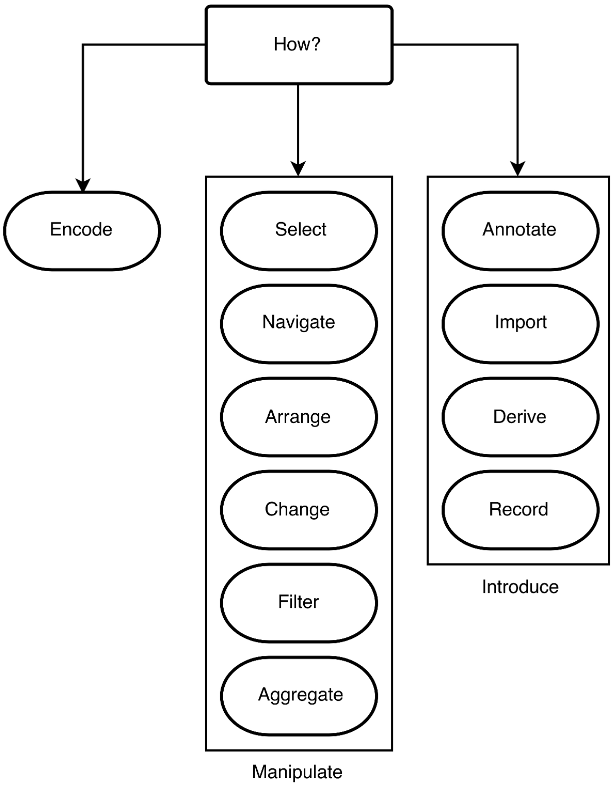


Figure 2.2: *How* tasks from Brehmer and Munzner's multi-level typology of abstract visualisation tasks [22].

with some of the tasks presented in Wehrend et al. [256], which include: retrieving values, filtering, computing derived values, finding extremum, sorting, determining ranges, characterising distributions, finding anomalies, clustering and correlating. The low-level tasks are based on student questions and potentially applying the same approach with professionals is likely to produce more questions.

Another taxonomy addresses the user's intent while interacting with the infovis system, and not with its sole focus on a user's goals, presents seven categories of interaction: select, explore, reconfigure, encode, abstract/elaborate, filter and connect [267]. This taxonomy is the result of a review of infovis taxonomy literature and infovis tools, as well as a survey of commercial infovis systems. This approach resulted in 311 interaction techniques, which are aggregated by the user's intent and grouped into different techniques using an affinity diagramming method.

How [22]	Amar et al. [5]	Yi et al. [267]	Heer et al. [92]
Encode		Encode	Visualise
Select		Select	Select
Navigate		Zoom/abstract	Navigate
Arrange	Sort	Reconfigure	Sort/organise
Change			
Filter	Filter	Filter	Filter
Aggregate			
Annotate		Add placemark	
Import			
Derive	Compute		Derive
Record		Redo/undo	Record

Table 2.8: A selection of taxonomies compared against Brehmer and Munzner's *how* class [22].

While the categories are not thoroughly inclusive of interaction techniques, the taxonomy provides a useful categorisation for understanding interaction when considering the user's intent.

Other taxonomies aim to address not only the user's intent but also the actions needed to affect the infovis system. For example, a categorisation of 12 analytical tasks are clustered into three high-leave groups: data and view specifications (visualise, filter, sort and derive), view manipulation (select, navigate, coordinate and organise), and process and provenance (record, annotate, share and guide) is proposed as a taxonomy of interactive dynamics for visual analysis [92]. This taxonomy addresses the user's intent and also provides example of actions to achieving this intent and their effect on the system's response.

A recent and particularly thorough classification is provided by Brehmer and Munzner [22]. The classification focused on *what* does the infovis task pertain, *why* the task is performed by the user and *how*. The typology is informed by taxonomy and typology literature, as well as works on user behaviour. The *why* class of the typology describes the user's intent and includes high-level to low-level intent: consume (present, discover and enjoy), search (lookup, locate, browse and explore) and query (identify, compare and summarise). The *what* class distinguishes between an infovis task's input and output.

From the typology, the *how* class is the class that this thesis is concerned with because it deals with the methods by which users interact (see Figure 2.2). The *how* class also better facilitates the communication between the user's intent and action, which is where TUIs benefit interaction. The class is subdivided into the following tasks:

- Encode
- Manipulate
  - Select: actions that distinguish between selected and unselected visual elements.
  - Navigate: actions that alter users' viewpoints.
  - Arrange: actions that spatially organise visualisations or visual elements.
  - Change: actions that alter the visual encoding.
  - Filter: actions that apply inclusion and exclusion criteria.
  - Aggregate: actions that alter the granularity of a visualisation.
- Introduce
  - Annotate: actions that add graphical or textual annotations to visual elements.
  - Import: actions that add new visual or data elements to a visualisation.
  - Derive: actions that compute new data for existing data elements.
  - Record: actions that capture persistent records of the visualisation.

Compared to other taxonomies (see Table 2.8), the tasks represented in the *how* class are more thorough and would better reflect infovis tasks covered by TUIs.

2.5.2 Infovis Interaction

Most infovis systems have WIMP interfaces that lack the advantages brought by interactivity beyond the traditional mouse, keyboard and desktop setups. Natural and efficient desktop interactions have been developed for manipulating infovis;

however, HCI has progressed towards interactions that are better at harnessing motor-cognitive abilities (e.g. touch [17, 200] and tangible interactions [191, 232]).

The term direct manipulation was first coined by Shneiderman with the introduction and proliferation of WIMP interfaces [219]. Direct manipulation is been described as, 'What you see is what you get', and aims to afford the user with the ability to see and naturally manipulate virtual content directly [221]. Benefits of direct manipulation include: learnability, efficiency, immediate system response, engagement and being more attractive. Five main principles characterise direct manipulation [219, 228]:

- Continuous representation of the objects of interest.
- The objects are manipulated with physical actions or labelled button presses instead of complex syntax.
- Rapid, incremental and reversible operations with immediate impact on the object of interest.
- Supports layered or spiral approach to learning that permits usage with minimal knowledge.
- Exploration without severe consequences.

Two underlying phenomena, distance and engagements, provide the user interaction with the feeling of direct manipulation from a cognitive perspective [102]. Distance refers to the distance between a user's thought and the system's physical requirements to accomplish that thought. The qualitative feeling of engagement relates to the conversational or model-world metaphors of directly manipulating an object of interest. These two dimensions work together to define the directness of a manipulation and interface.

WIMP interfaces are considered a form of direct manipulation; however, their directness is comparably low against touch and tangible interfaces and this introduces downsides that are alleviated with post-WIMP interfaces. In particular, metaphor confusion in WIMP interfaces as the complexity of the interface grows. For instance, a number of features can be easy to learn separately but when manipulated aggregately they are likely to cause confusion [254]. Post-WIMP interface styles such as TUIs hold the potential to lessen complexity since metaphors are controlled and contained within tangible objects that better utilise motor-cognitive abilities.

2.5.3 Infovis TUIs

There has been comparatively little research into the use of TUIs for data analysis, despite the fact that current visualisation systems could clearly benefit from more direct and flexible interfaces. Research in visualisation tends to overlap scivis and infovis, which are dissimilar. Scivis visualises the real-world accurately, e.g. molecules, while infovis visualises data that is abstract, e.g. genome networks.

There have been various applications of scivis TUIs. In scivis, the spatial representation is given and lends itself to physical representation. For instance, tangibles are autonomously used to control and contain projected data [187, 249]. Other applications combine tangible props of real-world scientific objects with various displays, e.g. a laptop [75], an interactive surface [205] or a vertical display [137]. Augmentation is often used with the tangible objects to project texture [137] or dynamic information [75]. Tangible controls are used to manipulate digital scivis on a desktop [96] and on interactive displays [10, 266]. Virtual reality (VR) is combined with tangible controls to manipulate virtually projected data [107, 135, 203].

The majority of infovis TUIs provide interactions covering the *how* part of Brehmer and Munzner's typology [22] (see Figure 2.2). A good proportion of these applications are developed for the purpose of exploring genomic information or querying databases, while others adopt tangibles to explore node-like structure. Popular techniques in infovis, e.g. lens-based interaction, are often adopted as well. The rest of this section surveys infovis TUIs while reflecting on the infovis tasks and interactions (see Table 2.9), size of the data and implementation technology (see Table 2.10).

2.5.3.1 Early Interfaces

One of the earliest examples of an infovis TUI is Urp, a system that combined the use of a workbench and physical architectural models for urban design and planning [250]. It utilised the I/O infrastructure [249] to simulate architectural

Paper	Manipulate						Introduce			
	Select	Navigate	Arrange	Change	Filter	Aggregate	Annotate	Import	Derive	Record
Early interfaces										
Ullmer et al. [242]		●		●						
Underkoffler et al. [250]	●		●	●		●				●
Jacob et al. [108]	●		●			●		●		●
Genome interfaces										
Shaer et al. [211]	◇	◇	◇	●◇	●	◇	◇	◇		◇
Shaer et al. [214]	◇	◇	◇	◇	◇	◇	◇	●◇		◇
Arif et al. [9]	○◇		●	●	●○	●				
Valdes et al. [253]⋆			●	●○	●○◇	●○				
Biology and Health interfaces										
Schneider et al. [204]			●			●◇				
Claes et al. [40]		○	●		○	●				
Tangible query interfaces										
Jetter et al. [111]	◇	◇	●		●◇	●				
Radle et al. [191]	◇	◇	●			●				●
Ullmer et al. [245]			●		●	●				
Klum et al. [133]			●		●	●				
Physical visualisation interfaces										
Taher et al. [232]⋆	○	○◇	◇		○◇					
Lens-based interfaces										
Koike et al. [134]†			●	●	●	●				

Continued on next page

Paper	Manipulate						Introduce			
	Select	Navigate	Arrange	Change	Filter	Aggregate	Annotate	Import	Derive	Record
Spindler et al. [227]†	○	●	●	●	●	●				●
Kim et al. [124]†	○●	○	●	●	●	●				
Ebert et al. [55]†		●◇	●	●	●◇	●				●◇
Geographic visualisation interfaces										
Nagel et al. [169]	◇	◇			●			●◇		
Dumas et al. [53]		●			●					
Ma et al. [149]					●			●◇		
Other interfaces										
Luderschmidt et al. [147]	●	◇	●◇		●	●◇	◇			●
This research										
eQTL infovis TUI	◇	◇	●	●	●	●		◇	●	

Table 2.9: TUIs' infovis tasks based on Brehmer and Munzner's *how* class [22]. Tasks performed with tangible objects are represented with ●. Touch manipulations on an interactive surface are represented with ◇, while touch actions on a tangible object are represented with ○. The table is only inclusive of tasks performed on infovis. Two papers (⋆) explored the interaction design space and thus provide a wider range of interactions. Other papers (†) explored case studies that were categorised as infovis.

Paper	Technology			Data set	
	RFID	Micro-controller	Computer vision	Type	Size
Early interfaces					
Ullmer et al. [242]			●	Planar	
Underkoffler et al. [250]			●	Planar	
Jacob et al. [108]	●		●	Temporal	
Genome interfaces					
Shaer et al. [211]			●	Multi-dimensional	
Shaer et al. [214]			●	Multi-dimensional	
Arif et al. [9]		●	●	Network	
Valdes et al. [253]		●	●	Multi-dimensional	
Biology and health interfaces					
Schneider et al. [204]			●	Hierarchical	
Claes et al. [40]		●		Infographic*	
Tangible query interfaces					
Jetter et al. [111]			●	Multi-dimensional Network	204 hotels
Radle et al. [191]			●	Multi-dimensional Network	7000 documents
Ullmer et al. [245]		●		Multi-dimensional	
Klum et al. [133]		●		Multi-dimensional	1500 electronic documents
Physical visualisation interfaces					
Taher et al. [232]		●	●	Multi-dimensional	10-83 rows

Continued on next page

Paper	Technology			Data set	
	RFID	Micro-controller	Computer vision	Type	Size
Lens-based interfaces					
Koike et al. [134]			•	Planar†	
Spindler et al. [227]			•	Multi-dimensional†	
Kim et al. [124]			•	Multi-dimensional†	
Ebert et al. [55]			•	Planar†	
Geographic visualisation interfaces					
Nagel et al. [169]			•	Planar	116 architectural projects
Dumas et al. [53]	•	•		Planar Temporal	28000 artworks
Ma et al. [149]			•	Planar	
Other interfaces					
Luderschmidt et al. [147]			•	Network	
This research					
eQTL infovis TUI			•	Multi-dimensional	230912 SNPs for each file

Table 2.10: Technologies and data sets used in infovis TUIs. (⋆) Infographics are not part of Shneiderman et al. [220] data types, but the presentation of the work is best described as infographic. (†) Papers that explored case studies that were categorised as infovis.

shadow, reflection, wind, proximities and visual space. The architectural models were physical representations of the actual architecture. Urban simulation objects (distance, wind, anemometer, material transformation and clock for shadow) either denoted their function with symbolic objects (wind and distance measuring tools), or were represented with abstract forms.

Informal evaluations by professionals found Urp [250] generally favourable. Academics thought the system was useful to easily explain concepts for urban design and planning, while practicing architects considered it an invaluable aid for client presentations. Others architects thought it useful to explain ideas to fellow or senior practitioners. The physicality and familiarity of the objects seemed to minimise the domain knowledge hurdle and made the system accessible to general users.

A tangible interface consisting of a senseboard [184], data pucks and command pucks was developed to manipulate discrete pieces of abstract information [108]. The TUI aimed to combine the benefits of using a GUI and physically manipulating paper when organising information. The system was described within the context of organising conference proceedings for ACM CHI 2001, where the senseboard was used to organise data pucks representing submitted papers. Each conference paper's details were projected onto a puck in a cell that could be seen, grabbed and moved around the senseboard. The physical representations of the pucks were exploited to express commands, such as view details, group/ungroup, type-in via keyboard or copy a data puck's original data. For example, when a group command puck was placed on the first data puck it 'swallowed' pucks aligned below it into a group.

To explore the possible benefits of tangible interaction in a simplified work schedule organisation task, the TUI [108] was compared against three other conditions: paper, reduced-senseboard and pen GUI. The results showed better performance for the TUI condition than either paper or GUI. The TUI was also subjectively preferred over the other three conditions. Compared to the paper condition, the TUI was believed to have preserved the fluidity of physical paper and maintained its 'tangible thinking' qualities.

2.5.3.2 Genome Interfaces

The nature of genomic information (scale, heterogeneity and diverse domains) presents challenges for the development of TUI applications; however, the nature of TUIs offers an opportunity to enhance learning and collaboration [213]. G-nome Surfer, an interactive surface exploration system, was intended to aid collaboration when exploring genomic data [215]. Users of the system could explore eukaryotic (G-nome Surfer 1.0 [211] and 2.0 [214]) and prokaryotic (G-nome Surfer Pro [215]) genomic data.

On the interactive surface users were able to navigate the genomic data, access heterogeneous data, perform a basic local alignment search (BLAST) for similarity between biological sequences and arrange and dock windows around the surface. Three G-nome Surfers were developed, but only the first two versions adopted tangible objects for some of their function. In G-nome Surface 1.0 [211], a tangible object was used to search similarities in BLAST that allowed for immediate and visible view changes. With the introduction of a contextual help tool in G-nome Surfer 2.0 [214], a tangible flashlight tool was introduced to display glossary definitions to encourage the discussion of these terms.

G-nome Surfer 2.0 [214] was comparatively evaluated against traditional and multi-mouse GUI setups to investigate the system's support for collaborative learning. The results showed reduced workload and stress level for the tabletop and multi-mouse conditions, but the tabletop condition proved superior as it encouraged more participation and reflection. Despite the encouraging results, the impact of tangibility was not investigated since the tangible aspect was removed for the sake of the evaluation.

Active tangibles are autonomous physical objects that have sensing and actuation capabilities. They are programmable and thus may be reconfigured over time to be dynamically modified. Alternatively, passive tangibles are those that unify physical and digital content to relate physical action to digital output. Passive tangibles are prominently used on interactive surfaces, while active tangibles are more common with other forms of TUIs.

Sparse tangible is a hybrid system that combined active tangibles and an interactive table for the collaborative exploration of gene and protein networks [9].

A linear navigation approach was used to explore networks within a particular organism or gene via touch vertical swipe gestures on the active tangible. Once a structure was selected, an active tangible was placed on the tabletop to display all available networks virtually. Queries were constructed by stacking active tangibles one on top of the other.

The TUI [9] was evaluated by three domain experts that found the system to be fun, useful and easy to use. Several comments for improvement were also provided: expanding information and filtering options for the constructed queries. The use of active tangibles expand the space for interaction and system complexity, which is a somewhat drastic move from the immature tools typically used in genomic domains that could adversely affect the utilisation of the interface.

The gesture space of active tokens for manipulating large data sets on a tabletop was explored via a user elicitation study [253]. The study was conducted with six active tokens and either a horizontal or vertical multi-touch surface to complete a query-building task of personal genomics. To complete the task, participants manipulated both discrete and continuous query parameters. Using a think-aloud protocol, interaction logs and video recordings, gesture vocabularies were generated for eight commands.

The gestures generated involved: placing or hovering the token on the surface, tapping on a token and tilting a token or neighbouring tokens [253]. The results implicated design considerations for active tokens and interactive surfaces: continuous interaction, interaction beyond the surface and rescue of gestures. To elicit feedback, users' behaviours were always accepted by the system and thus not reflective of a continuous dialogue. The minimal affordance of the tokens also made it difficult to expand on new interaction ideas.

2.5.3.3 Biology and Health Interfaces

The size of the explored data sets decreased as the applications move on to more general biological and health domains. Phylo-Genie was developed on a tabletop to explore its effectiveness in fostering collaborative learning of phylogenetics [204]. Given a learning scenario, users manipulated a series of physical tokens that represented specimens from a scenario. Each token was in effect a container of a specimen's characteristics and appearance that was displayed when it was placed on the tabletop. The tokens were used to build phylogenetic tress by extending a given template where users reflected on and interpreted the structure of the tree.

The interactive surface [204] was comparatively evaluated against a typical pen and paper approach for collaborative learning support. The tabletop condition was found superior to the traditional condition in learning performance and engagement. In the collaborative task, the tokens seemed to enforce ownership of objects and areas on the interactive surface. Due to that, users collaborated with a balanced division of work, i.e. turn taking, instead of working independently. While learning performances were improved, this may largely be the result of the novelty effect of the new interface. The interactive surface was also found to be more physically demanding than the paper and pen condition.

Active tangibles were utilised in a casual health infovis prototype that aimed to engage layman users with their health [40]. It consisted of a set of three Sifteo cubes [164] representing data categories. The cube displays were manipulated via touch to access the category's data dimension. Sequential queries were formulated by connecting the displays to each other, where the dimensions' filters were combined to reveal a percentile value. The sides of each sifteo cube represented a specific data dimensional filter, and the cube was rotated in order to select a filter.

A lab study compared the prototype [40] against a GUI on an insight scale. Insight could either be factual, interpretive or reflective based on the depth of the insight. The majority of the GUI insights were found to be factual, while the TUI insights were more reflective. An in-the-wild study set up the active tangible and the GUI in a hospital waiting room. Users found the TUI more inviting as 18 users interacted compared to one user in the GUI. Similar to the results of the first experiment, the insights were mostly interpretative or reflective for the TUI. The TUI also encouraged social interaction, as well as the physical sharing of tangibles. The current prototype is restricted to three cubes and eight dimensions, which poses the question of this approach's suitability for exploring larger data sets. Also, the active tangible's screen size makes it impossible to incorporate plots when used autonomously.

2.5.3.4 Tangible Query Interfaces

Numerous tangible systems have explored the use of passive tangible and active tangible for constructing queries. Facet-streams was developed as a hybrid interactive surface for collaborative faceted product search [111]. It combined infovis techniques with tangible and touch interactions. Circular glass discs, facet tokens, enabled users to specify a set of personal criteria for narrowing down a search. A criterion was formulated by selecting a facet token and specifying a value via touch. By harnessing Boolean logic, a network of facet tokens can be connected to each other via streams to chains of criteria. The visual streams reflected direction and the number of results flowing through a stream. The streams also had an option to preview the results.

Two user studies were conducted to evaluate facet-streams' [111] collaborative use and the comprehensibility of the Boolean logic metaphor for filter/flow. The first experiment compared the hybrid interface against a web interface, where both interfaces were found to be effective for the collaborative task, with the former supporting different search strategies. Usability flaws were noted during browsing, mainly occlusion and combined orientations that led users to congregate at a single side of the tabletop. The second experiment evaluated users comprehensibility of the Boolean logic metaphor and found it to be learnable.

While the previous system dealt with a relatively small data set, 204 hotels [111], a collaborative and spontaneous search of a visualised online book repository system explored a database of more than seven thousand documents [191]. The TUI combined horizontal and vertical displays with tangible objects. Paper strips were used to formulate search requests and, when placed on the table, the system queried the repository for matches. Using a pipe and filter metaphor [111], paper strips could be concatenated with AND and OR operations. Result tokens were used to visualise search results on the vertical display in a scatter plot. Search results were compared by using multiple result tokens on multiple edges of the query network. To encourage serendipitous discoveries, similarity is computed based on author, title and abstract relations and the results shown. However, the paper does not offer comparative evaluation of the system against less elaborate setups.

A query interface, TQI, explored the use of tangible objects, active and passive, that represent database parameters in two interfaces [245]. The first TQI consisted of a display surface, a query rack, and a number of parameter wheels representing database fields. The display surface showed two visualisations, geographical and scatterplot views, that were updated by manipulating the parameter wheels docked into a query pad (token+constraint system [246]). Queries were constructed by bringing two query pads together, which performed an AND operation to the parameters. The second interface expressed queries with parameter bars and a GUI monitor. While parameter wheels were passive, bars were active tangibles with a display and double sliders to manipulate its bounds. Similar to the first interface, parameter bars were bought together to form AND operations, as well as forming OR operations by spatially separating the bars.

A preliminary user study compared the parameter wheels TQI [245] with a GUI query interface. Users took longer to complete a query task, albeit not significantly, with the TUI due to an additional setup requirement. User preferences were split between the TUI and the GUI, while the TUI was more likely to support effectiveness. Two-handed interactions were also observed during the evaluation and it was found that 80% of users used both hands with the TUI. 40% of the users also recalled that using two hands while manipulating the wheels was one of the major strengths of the TQI system. Despite the results, it is claimed that this form of physical representation could scale to larger data sets.

Stackables [133] are tangible widgets developed for faceted browsing, where users could search, share and manipulate browsing search results. The tangible system consisted of stackables (facet token and ground plates) and an output display that showed the results of the queries. Facet tokens were rectangular-shaped active displays showing the categorical facet values that could be manipulated to build up a query. Ground plates formed the bases of a stack of facet tokens. Facet values were navigated with wheels located on both ends of the display, and selections were made with a press of a button. Queries were constructed by stacking a number of facet tokens one on top of the other and results were shown on the output display.

The system [133] was evaluated with a data set of 1,500 books and nine facets in several tasks that examined the usability of the selection wheels, result representation, range selection and query building. The TUI's concept was validated for faceted information search, where the stacking metaphor was immediately understood by the majority of the users. All users could see themselves using stackables in a group scenario. The evaluation also exposed concerns about the visualisation used to represent the faceted query results. The bulkiness of the facet object could cause fatigue and was somewhat clumsy to handle, and, unlike previously described active tangibles, they did not support touch.

2.5.3.5 Physical Visualisation Interfaces

Physical visualisation is defined as, '[...] visualisations that are made of physical matter, as opposed to presented on a computer screen or projected on a surface as it is traditionally the case. This includes matter whose shape or properties change over time' [110]. The physicalisation of visualisation is a relatively new approach to representing infovis, which involves turning a visualisation into a physical information display. Current technology utilised for the physicalisation of visualisation is still in its infancy, which limits the size of the data and range of interactions [109].

EMERGE is a physical dynamic system developed for visualising bar charts [232]. The system design followed from inFORM [66] but was adjusted to enhance the interaction space. The system consisted of 10x10 rods that are manipulated to achieve data analysis tasks. It could sense touch and gesture using a mounted projector and Microsoft Kinect. Interaction techniques were developed for common tasks, such as filtering, organisation and navigation. The techniques involved a combination of approaches varying from touch and gesture input to the physical manipulation of rods.

The shape changing display [232] was evaluated to elicit initial thoughts and feedback, as well as reactions to the various interaction techniques designed for each task. Physical interaction with the rods was positively received and found to be intuitive and informative. The evaluation also provided insight on interaction modalities, learned behaviours and user reactions from the physical model. While physically interacting with infovis may better harness motor-cognitive activities, their rigidness restricts not only the size of the data set being visualised but also the ability to swap and change visualisation views.

2.5.3.6 Lens-based Interfaces

Geospace is a prototype application on the metaDESK that utilised the concept of using an active physical lens as a portal into digital space [242]. Geospace consisted of several tangible objects that were used to interact with a geographical space on the metaDESK surface. A physical object shaped as a great dome building acted as a container for the digital information about a campus. When the dome was placed on the surface 2D and 3D campus maps were activated, the former displayed on the surface and the latter on an active lens display. The physical object was used to navigate the maps, and when combined with another object the map could be scaled and rotated. Physical lens were also implemented to overlay the map with another view.

Lenses have also been commonly used in infovis to interactively parameterise spatial selections that alter global visualisations [239]. A number of applications had been developed to explore tangible lenses for infovis. Magic Lenses are virtual lenses used in a GUI environment to support the focus+context paradigm [19]. A tangible transparent version was later developed on a liquid crystal display (LCD) to allow for the detection of transparent 2D markers [134]. This avoids the black and white patterned fiducial makers typical to computer vision systems.

A number of case studies were developed to explore the use of tangible Magic Lenses [134], such as geographic visualisations. In the geographic case study, the user could change, filter or aggregate virtual content by interacting with the Magic Lenses. For instance, a Google Maps satellite image was changed to a normal map by placing the lens on the surface. In another example, a user overlapped two lenses to aggregate filters. While the transparent markers are generally non-obtrusive, their transparency made them harder to detect, leading to accidental obstruction of the markers.

Tangible views are displays that were used in conjunction with a tabletop to enhance common interactions in infovis [225, 226, 227]. A tangible view is lightweight and spatially aware in a way that it can be used as a local display or an input device when moved on or above a tabletop. As a tool for representation, tangible views along with the tabletop related to the focus+context concept of interaction and could also be used as a toolbox of tools. As a tool for interaction, it utilised the space surrounding the tabletop to perform common interaction techniques in the domain of infovis such as translation, rotation, direct pointing and gestures.

An example case study [227], illustrated the use of tangible views as a fisheye lens to control the location and magnification of scatter plot data via translation and rotation. Initial feedback from example case studies found the interaction natural to use. While handling tangibles over the tabletop expanded the design options and possibly allowed for more intuitive interaction techniques, handling the tangible for too long may prove fatiguing unless used infrequently.

Embodied lenses used tagged sheets of normal paper or transparent foil to further extend the physical metaphor of lenses (e.g.[227]), which allowed users to overlap lenses to control object composition [124]. An embodied lens was placed on a tabletop and registered to change the global visualisation within the lens. Lenses were moved and rotated to control the focal visualisation. When two embodied lenses overlapped, the lens regions were composed.

Time performances were compared for embodied [124] and virtual lenses in a canonical visual query task. Results showed that there was no significant difference in time performance, but it appeared that the embodied lenses promoted faster task performance largely due to using two-hands and eyes-free manipulation. While the experiment was not focused on infovis, the research provided application examples for multi-dimensional data visualisation, such as scatter and parallel coordinate plots and map interaction.

TangibleRings are hollow circular tangibles used on an interactive display that aim to overcome the problems caused by opaque or translucent tangibles that occlude and block interaction with occluded content [55]. TangibleRings could be used to navigate, select and aggregate composites by nesting the hollow rings. A sample map-based scenario was used to exemplify the possible manipulations to manage different information layers and filters.

Each tangible ring [55] filtered the map view based on the ring's defined function. The rings also offered individual views that could be adjusted via rotation or touch interaction, saved and shared with other collaborators, as well as expanded to form a new global visualisation. In order to allow interaction within a ring, the size would have to be large enough and the borders shorter. This in turn would have an effect on the number of rings that could be maintained on the interactive surface. Evaluation were not carried out to determine the effectiveness of the nesting approach.

2.5.3.7 Geographic Visualisation Interfaces

Venice Unfolding is a geo-visualisation TUI that was used to query an architectural project in Venice [169]. The system combined an interactive surface with a single asymmetrical polyhedron object, where each one of the object's literal facets (excluding the base) was used to represent a collection of query facets. Since each side of the polyhedron represented a collection of facets, a user would tilt the object and place it on the surface to display the facets. The polyhedron was rotated to browse the facet values, which were selected by pushing the objects towards an option.

A formative user study was conducted to gather initial feedback about Venice Unfolding [169] with simple exploratory tasks and a post-test questionnaire. Users interpreted the visualisation correctly as they were familiar; however, the awkwardness of the object shape was not immediately understood. Some users tried to use the passive objects interactively, while others did not associate the task with the interaction technique. These results highlight the importance of balancing between objects, shapes and interactions; unfamiliar object shapes are tricky to handle and mapping interaction may prove difficult to design and understand.

ArtVis is a tangible interface that combined advanced visualisation techniques and tangible interactions to explore large collections of artwork geographically and temporally [53]. The TUI represented a painter's working environment with tangibles such as a painter's easel, tube box and palettes. The ArtVis application was navigated using a collection of USB-controlled play-and-play components: a joystick and rotation sensors panned and zoomed, a slider navigated the artwork temporally, and an RFID reader scanned various tagged objects that represented different types of artwork.

User experiences with ArtVis [53] were evaluated in an informal evaluation where users were asked to explore the artwork data set. Users were observed during their interactions. They were also asked to fill out a standardised user experience questionnaire and interviewed at the end of the evaluation. Based on the questionnaire, ArtVis was found to be attractive and stimulated users, which was also observed by the researchers as the users kept interacting with the system longer than they had to. Users seemed to also use all the various objects provided with the TUI. While the system utilised various technologies with high-specificity objects, the clutter and the distant embodiment may prove cumbersome for interaction.

A museum exhibit enabled users to explore the distribution of oceanic phytoplankton using three ring objects and a 55-inch interactive table [149]. The visualisation showed the distributions of four types of phytoplankton in the oceans over time, and each phytoplankton was represented in a different colour pattern. A user would place a ring object on a location in the ocean to display the morphology of the different planktons. To access a textual guide that describes the different phytoplankton, the user would touch the tab on the side of the lens. A pure touch version of the system was also developed in order to comparatively assess the performance of both interfaces.

The tangible system [149] was compared against a touch version to understand the strengths and limitations of TUI versus touch. Visitors to the museum exhibit were observed and videotaped during their interactions with the systems. A think-aloud protocol was also applied for some of the visitors. The results of the study showed that the physical rings encouraged initial interactions and continued engagement. The physical rings also attracted more group interactions than their touch counterpart, arguably due to their visibility. Because of the TUI's setting in a museum exhibit, only limited functionality was provided.

2.5.3.8 Other Interfaces

A TUI has also been used to visualise information in the vicinity of safety critical information. Vispol (visualisation for the police) is a graph-based visualisation used on a tabletop, where multi-touch gesture and tangible objects are employed for input [147]. Incorporating existing work practices in a police station, the system used node-like diagram visualisations to analyse the topology of criminal networks. The diagram visualised persons or objects as nodes, and their connections to each other. Each node and connection was further identified with symbolic attributes, as well as free-text entries.

Vispol [147] could be interacted with via direct touch and a number of abstract person, timestamp, magnet and layout tangibles. Nodes in a graph could be edited and arranged, and connections could formed via touch gestures. Person tangibles were placed on a tabletop to select nodes, or rotated to open a node's configuration dialogue. Connections were formed by bringing person tangibles together. Timestamp tangibles were used to save or reopen a configuration by placing them on the surface. The visualisation was filtered with magnet tangibles that attracted notes that meet their filtering criteria. Node organisation was also automated using layout tangibles. While the node-like visualisation was met positively by the police officers in an informal evaluation, interactions with the tabletop via touch and tangibles were hesitant and cautious.

2.5.4 Summary

Visualisations help users make sense of large data sets in order to detect patterns, discern relationships and to check the data for anomalies. Often, the term is used to represent various forms of visualisations, mainly: scivis and infovis. Due to the abstract nature of the case study (see Chapter 3), this research focused on infovis. There were several systems developed as infovis TUI, however the number is low compared to other application domains. The surveyed infovis TUIs were summarised based on technology and data type (see Table 2.10) and their interactions with the infovis classified using Brehmer and Munzner's *how* class [22] (see Table 2.9). The infovis TUIs surveyed enabled this research to learn from previous implementation, applications and interactions.

Computer vision was the most common technology adopted, while micro-controllers were sometimes utilised and combined with either interactive displays [9, 253] or passive displays [53, 133]. Multi-dimensional, planar and network data types were typically explored by a large proportion of the TUIs. Data size ranged from very little to genomic proportions. For larger data sets (50,000 data items or more) an interactive surface was typically utilised with passive tangibles [204, 211, 214] and only occasionally with active tangibles [9]. Several of the interfaces balanced between touch and tangible interactions for the task [55, 111, 147, 204]. Due to the nature of infovis, the majority of visualisations were represented digitally and manipulated using tangibles [9, 55, 111, 124, 133, 134, 147, 149, 169, 191, 204, 211, 214, 227, 253]. Improved effectiveness and efficiency were commonly claimed by the studies; however, the majority did not conduct experiments to evaluate user performance (see Section 2.6.3).

2.6 Evaluation

Evaluation is the process of collecting data about current practices or new systems from interested participants within a specified context [189]. Evaluations have been carried out in this thesis to elicit requirements and comparatively assess function. Various methodologies are considered based on their strengths and limitations and their potential influence on the findings. In this section, common evaluation methods in HCI are described and include: observations, interviews, focus groups, questionnaires, case studies, usability studies, heuristics, controlled experiments and logs. Lam et al. [139] identified seven infovis evaluation scenarios conducted at various stages of a project's lifecycle, which is described next. The scenarios and corresponding techniques are then surveyed within the context of infovis TUI systems and how they relate to the evaluation methods adopted in this research.

2.6.1 HCI Evaluation Methodologies

There are various kinds of system evaluations in HCI. Analytical evaluations reason through direct analysis, while empirical evaluations make observations or measurements. Evaluations can also be applied earlier in a project's lifecycle to evaluate and refine ideas (formative) or later in the lifecycle to evaluate systems (summative). Data collected via evaluations can be qualitative or quantitative in nature. Qualitative data are commonly in the form of text, images or analogue data collected using protocols or open-ended questions from a small sample size. Data collected from a large sample group are typically quantitative and numeric.

The rest of this section describes common evaluation methods used in HCI while stating the strengths and weaknesses of their application in research.

2.6.1.1 Observations

Observation is the process by which researchers observe participants in a field or laboratory study to understand how system features influence the use of technology. Observational data are mainly collected to understand relatively unexplored work practices or to elicit information that explain behaviour in a particular setting [138]. Field observation is the careful observation of users in a real-world setting as they perform current practices or interact with new systems. When more control is required in the study, laboratory observations can be carried out in a laboratory setting [139].

Complex and rapidly changing events require frameworks to structure and focus observation. Some frameworks simply focus on the user, place and thing, while others provide more detailed frameworks that focus on particular situations [189]. Observational data are typically collected via video or audio recordings that are transcribed and then analysed. Transcription can take hours to process and is analysed to represent themes, patterns and stories [138].

2.6.1.2 Interviews

Interviews and focus groups involve feedback from individuals that are of direct interest to the research. Interviews are typically run with one individual at a time, while focus groups involve multiple users at the same time. Of the two, interviews are more labour intensive as they require individual meetings. The application of interviews and focus groups in HCI helps researchers prior to development, during the development process and to summatively evaluate a final product. Interviews and focus groups have been used in HCI as a means of initial exploration, requirement gathering and evaluation [48].

Feedback from interested individuals is elicited in interviews and focus groups with questions that can be structured, semi-structured or unstructured. Structured questions limit participants to a small number of questions with preset choices (e.g. Likert scale questions). Semi-structured questions are more loosely structured and the interviewer is likely to use a guide to adhere to questions and topics that need to be covered. Interviews and focus groups with unstructured questions follow a focused goal but do not follow a set of themes or structures. Structured questions are the easiest to analyse but are likely to discourage elaboration. Due to the loose structure and open-endedness of semi-structured and unstructured questions, conversation is often stimulated and deep insight generated. However, unstructured questions can fail to identify points of comparison between various interviews during analysis.

Interviews and focus groups are often analysed using qualitative data analysis methods to identify common ideas that arise during questioning. Content analysis examines patterns in the interview text by analysing structural markers, while discourse analysis considers the overall structure of the text. User responses can also be categorised systematically or interpretively to organise important concepts and the relationships between them. Critical-incident analysis identifies incidental stories, i.e. case studies, that can provide useful information [140].

2.6.1.3 Questionnaires

Due to ease of administration and analysis, questionnaires are the most commonly used research method in various disciplines. Despite the popularity of questionnaires, the construction of the questions is no easy feat and should be developed in a way that avoids leading, biased questions and ambiguous responses [179]. There are many existing questionnaires that have been rigorously developed and validated in HCI literature, e.g. questionnaires for user interaction satisfaction (QUIS) [37] and usefulness, satisfaction, and ease of use questionnaires (USE) [148].

Questionnaire questions can either be open-ended or closed-ended. Open-ended questions offer no predefined options to choose from, while closed-ended questions limit the answers to a set of response options. One of the main concerns of closed-ended questions is the bias that may result from suggesting responses. Open-ended questions fall at the risk of imposing broad questions that are difficult to respond to. Questionnaire analysis depends on the types of questions used and number of responses gathered. Typically, the data is cleaned first by validating responses and questions are categorised as quantitative or qualitative. Quantitative data are statistically analysed descriptively or analytically. Qualitative data is first coded whereby concepts are then categorised to form a theory [179].

2.6.1.4 Case Studies

Case studies involve the in-depth examination of a specific instance or multiple instances within a specific context using a small number of interested individuals to gather requirements and evaluate prototypes and final products. Case studies are commonly adopted to gain descriptive or explanatory understanding about specific situations, explore novel situations or demonstrate a newly developed tool. A case study could either address single or multiple instances at a holistic level or with embedded cases with an overall holistic approach. Conducting multiple cases at the same time is typically more difficult to implement but ensures greater confidence in the findings [269].

Once a design is chosen for a particular case study, data is collected by various means that address different perspectives, which include: interviews, observations, archives and documentation. A protocol is then developed to determine how and what data will be collected. The analysis and interpretation of a case study depends on the data collection methods used and are generally qualitative in nature. Common analysis techniques include: pattern matching, explanation building or chronological analysis [269].

2.6.1.5 Usability Testing and Heuristics

In usability testing, researchers examine interested participants as they interact with representative tasks in representative environments [143]. This method is commonly adopted to test iterative prototypes from low-fidelity prototypes to working versions of software prior to a system's release. Data is often collected in a controlled laboratory environment using video recordings and interaction logs. The collected data is compiled to determine usability issues and provide developmental refinements to ensure the satisfaction of usability requirements [189].

Heuristics are a set of usability principles used by researchers to inspect a system for usability problems in a heuristics evaluation [175]. Heuristics evaluations are commonly applied to identify usability problems in an iterative design process.

Expert usability evaluators inspect the system and data is collected and aggregated from all evaluators. This form of evaluation obtains feedback quickly early in the design process and relatively inexpensively. However, usability experts with domain expertise are hard to come by and may prove expensive.

There is a large body of literature that identifies heuristics and explains usability problems. Heuristics developed by Nielsen et al. [173] are one of the well-known usability principles synthesised from usability problems collected from previous projects. The heuristics were categorised into ten top factors that include: visibility of system status, match between system and real world, user control and freedom, consistency and standards, error prevention, recognition rather than recall, flexibility and efficiency of use, help users recognise, diagnose and recover from errors, help and documentation [173, 174].

2.6.1.6 Controlled Experiments and Logs

A controlled experiment is a traditional scientific method adopted from psychology that is widely used in HCI to evaluate systems and interactions. It is largely summative and quantitative for the purpose of comparatively assessing conditions. In some instances, it is used formatively to isolate and assess important factors of a system. A controlled experiment is designed around a testable hypothesis that is concerned with a task, feature, measurement and population, from which dependent and independent variables are determined, as well as conditions, population and arrangement. An experimental protocol is then designed to satisfy external and internal validity [29].

Controlled experiments are mainly concerned with quantitative data that are collected using automatic logs, video or audio recordings, questionnaires and interviews. Almost all controlled experiments are analysed through a significance test (descriptive or analytical) that is decided on during the design process. Conclusions are then drawn from these tests and reported. A common structure for reporting includes: method (participants, materials and procedure), results and discussion [140].

Logs are automatic records of users' interactions with a system, such as key presses and button clicks, that allow sessions to be recreated. Performance data can also be captured, such as time and errors. Logs are mainly quantitative and typically self-contained. Researchers analyse logs to draw conclusions about system usage or behaviours. This research method is commonly used as it can easily be conducted and evaluated. Nevertheless, logs are error-prone unless designed properly and piloted to record realistic data [189].

2.6.1.7 Summary

Evaluation methods in HCI are usually combined and performed at numerous points of the development lifecycle to achieve thorough analysis of design needs, usability and performance. Observations allow researchers to view what users do in context, but can be obtrusive and analysis can be time consuming. Interviews and focus groups are used by researchers to discover ideas and understand opinions from a small sample of users. Similar to observations they are relatively difficult to conduct and analyse. Also addressing a limited sample, case studies may discover individualistic results that are not representative. Questionnaires can address a larger sample group compared to other methods, but when poorly designed can risk bias. Usability testing identifies usability problems following a set of guidelines, while heuristics are general rules of thumb. Both methods are relatively inexpensive and easy to conduct. However, heuristic evaluation can prove expensive when usability experts are hard to find. Controlled experiments commonly utilise various methods of research, such as logs, to systematically assess a system's overall performance. The design and running of a controlled experiment can be both expensive and time consuming.

2.6.2 Infovis Evaluation

Since 2000, with a special issue of the International Journal of Human-Computer Studies [13], there has been a call for a shift towards considering the usability and conducting evaluations of infovis tools and techniques. Up to then, visualisation methods were often presented without any report or evaluation of their usability.

An analysis of 850 empirical studies (361 with evaluations) identified seven evaluation scenarios spanning all stages of a project lifecycle for understanding data analysis and visualisation [139]:

- Understanding environments and work practices (UWP)
- Evaluating visual data analysis and reasoning (VDAR)
- Evaluating communication through visualisation (CTV)
- Evaluating collaborative data analysis (CDA)
- Evaluating user performance (UP)
- Evaluating user experience (UE)
- Evaluating visualisation algorithms (VA)

The first four scenarios are for understanding the analytical process to elicit requirements, assess a tool's support for the process, learning, sharing and collaborative work. UWP evaluations elicit formal requirements and design implications for an infovis by understanding current work practices with traditional software or without. This is achieved via field or laboratory observation and interviews. Evaluations within VDAR aim to asses a visualisation tool's support for the analytical process using case studies on domain experts, controlled experiments and laboratory-based observations and interviews. CTV evaluations use controlled experiments or field observation and interviews to asses a visualisation tool's ability to support learning and sharing. To assess a visualisation tool's ability to support collaboration, a CDA evaluation is applied using heuristic evaluation, log analysis or observation in the field or laboratory.

The last three scenarios are used to assess objective performance, subjective experience and the quality of a visualisation algorithm. UP evaluations apply controlled experiments or field logs to assess the user's performance when using a visualisation tool. Evaluations within UE aim to understand users' thoughts and subjective experiences when using a visualisation through informal evaluations, usability tests, field observations and questionnaires. The final scenario, VA, considers the performance quality of a visualisation algorithm using two main classes of methods: visualisation quality assessment and algorithmic performance. Table 2.11 summarises the seven scenarios and the methods applied within each scenario as surveyed by Lam et al. [139].

The methods iterated in Table 2.11 largely encompass the common methodologies used in HCI research (see Section 2.6.1). Methods not described earlier include: informal evaluation, visualisation quality assessment and algorithmic

Scenario	Evaluation methods
Understanding environments and work practices (UWP)	Field observation Interviews Laboratory observations
Evaluating visual data analysis and reasoning (VDAR)	Case studies Laboratory observation and interviews Controlled experiments
Evaluating communication through visualisation (CTV)	Controlled experiments Field observation and interviews
Evaluating collaborative data analysis (CDA)	Heuristic evaluation Log analysis Field or laboratory observations
Evaluating user performance (UP)	Controlled experiments Field logs
Evaluating user experience (UE)	Informal evaluation Usability test Field observation Laboratory questionnaire
Evaluating visualisation algorithms (VA)	Visualisation quality assessment Algorithmic performance

Table 2.11: Seven infovis evaluation scenarios and the methods applied for each scenario as surveyed by Lam et al. [139].

performance. The term informal evaluation is used in HCI as an umbrella term that includes heuristic evaluations and usability testing. Nevertheless, in infovis literature, informal evaluations are qualitative evaluations that are commonly carried out with domain experts in an informal setting. Tasks are not set and experts are encouraged to try out the system.

Visualisation quality assessment and algorithm methods are used for the evaluation of visualisation algorithms. Visualisation quality assessment automates procedures that compare generated solutions with the visualisation algorithm's goal based on image quality measures. Algorithmic performance evaluates the efficiency of an algorithm depending on its resource usage and visual output. Typically, algorithmic performance involves comparing the algorithm against others based on quantitative parameters [139].

2.6.3 Infovis TUI Evaluation Methodologies

In a survey of 22 infovis TUI systems, under six of the seven scenarios [139] the following methods were applied (see Table 2.12):

- UWP: interviews, field observation and questionnaires
- VDAR: controlled experiment and case studies
- CTV: controlled experiment
- CDA: laboratory observations and log analysis
- UP: controlled experiment
- UE: informal evaluation, laboratory questionnaire, interviews, usability test and field and lab observations

VA evaluations were not carried out in the literature. The majority of the studies combined one or more evaluation techniques at various stages of development. The rest of this section summarises the methods applied in the surveyed infovi TUI literature.

UWP Interviews are open-ended and exploratory techniques are used to form an understanding of the practices and needs of the potential users of a developed visualisation tool or system. Interviews are commonly used as a means for initial exploration or to elicit and gather requirements. Interviews can be fully structured, unstructured and semi-structured. For the design of Phylo-Genie, interviews were conducted with evolutionary biology experts to identify elements that support learning engagement with phylogenetic [204]. Semi-structured interviews were conducted with molecular and computational biologists to understand work practices and techniques in order to develop G-nome Surfer [211, 214]. To gather requirements for a police crisis incident, a series of interviews were conducted with police officers to develop Vispol [147]. For a lens-based TUI, interviews were conducted to derive users' conceptual model on spatial filtering using physical lenses [124].

Paper	Understanding data analysis				Understanding visualisation	
	UWP	VDAR	CTV	CDA	UP	UE
Early interfaces						
Ullmer et al. [242]						
Underkoffler et al. [250]						Informal evaluation
Jacob et al. [108]					Controlled experiment	Questionnaire
Genome interfaces						
Shaer et al. [211]	Interviews			Lab observations		Usability test Questionnaire
Shaer et al. [214]	Interviews		Controlled experiment	Lab observations	Controlled experiment	Questionnaire
Arif et al. [9]						Informal evaluation
Valdes et al. [253]						Usability test Lab observations*
Biology and health interfaces						
Schneider et al. [204]	Interviews			Lab observations	Controlled experiment	Questionnaire
Claes et al. [40]		Controlled experiment				Field observation
Tangible query interfaces						

Continued on next page

Paper	Understanding data analysis				Understanding visualisation	
	UWP	VDAR	CTV	CDA	UP	UE
Jetter et al. [111]		Controlled experiment		Lab observations Log analysis		
Radle et al. [191]	Questionnaire⋆					
Ullmer et al. [245]						Usability test Questionnaire
Klum et al. [133]		Controlled experiment				
Physical visualisation interfaces						
Taher et al. [232]						Usability test Questionnaire
Lens-based interfaces						
Koike et al. [134]		Case studies				Informal evaluation
Spindler et al. [227]		Case studies				
Kim et al. [124]	Interviews	Case studies			Controlled experiment	
Ebert et al. [55]		Case studies				
Geographic visualisation interfaces						
Nagel et al. [169]						Usability test Questionnaire

Continued on next page

Paper	Understanding data analysis				Understanding visualisation	
	UWP	VDAR	CTV	CDA	UP	UE
Dumas et al. [53]						Usability test Questionnaire Lab observations⋆ Interviews⋆
Ma et al. [149]						Field observation
Other interfaces						
Luderschmidt et al. [147]	Field observation Interviews					Informal evaluation
This research						
eQTL infovis TUI	Interviews[1]				Controlled experiment[2]	Informal evaluation[3] Questionnaire[4] Lab observations[4]

Table 2.12: Evaluation scenarios and method [139] that were applied in the infovis TUIs. (⋆) Methods that were not reported by Lam et al. [139] under that scenario (see Table 2.11), but were carried out in the study. (1) See Section 6.4.12. (2) See Section 4.4 and Chapter 7. (3) See Section 4.2. (4) See Chapter 7.

Field observation is the attentive observation of domain users to capture processes and understand work practices. Field observations occur in real-word settings where users interact with their systems freely. Vispol employed field observations (along with interviews) by visiting a control room in a police station used for the management of a crisis situation to develop Vispol's system requirements [147]. A questionnaires (while not inclusive under UWP) was used to inform a query interface [191]. The questionnaire was analysed by clustering similar needs and requirements of a query interface for physical and digital libraries.

VDAR For the purpose of assessing a visualisation tool for its support for the analysis process, controlled experiments and case studies are the methods utilised in the infovis TUI literature. Controlled experiments use laboratory experimentation to isolate and investigate important factors of a visualisation tool. For a TUI for casual health visualisation, insight generation was evaluated in a comparative experiment between the developed TUI and GUI, where the TUI was found to evoke more reflective insights [40]. For the purpose of evaluating design concepts, controlled experimentation was employed to study query formulation in a tangible query interface, Stackables [133]. Controlled experiments were also used for evaluating facet-streams and the comprehensibility of an applied metaphor [111]. Case studies are primarily used to understand a visualisation's tool support for tasks and processes. From the literature, lens-based interfaces [55, 124, 134, 227] used case studies to illustrate the tools support for infovis.

CTV G-nome Surfer 2.0 [214] employed controlled experiments to evaluate the user's obtainment of learning goals and level of participation. Nevertheless, the study was applied after the removal of contextual help, which was the only task requiring the use of a tangible object. Therefore, the evaluation was carried out on a multi-touch application and not a TUI.

CDA Laboratory observation can be used to evaluate a visualisation tool's support for collaborative analysis and decision making. This method was used for evaluating the user's collaboration on G-nome Surfer 1.0 [211], where observations were recorded as well as having the user provide a score of the extent of collaboration. Laboratory observations were also applied for the comparative evaluation of facet-streams [111] against a web-based version in a collaborative search task. The observations were combined with user activity traces and log information in order to combine clear assessments of interaction from observation with easily evaluated interaction.

A set of collaboration profiles were proposed by Shaer et al. [214] to understand the process of collaboration and to identify a profile's effect on the efficiency and effectiveness of a collaborative visualisation tool. The collaboration profiles were used in the evaluation of Phylo-Genie [204] in a comparative experiment against a paper-based approach traditionally adopted for learning phylogenetic. From laboratory observations, the collaborative profiles were identified and the tabletop approach was found to support significantly more collaboration than the paper-based approach.

UP Controlled experiments was the method of choice for evaluating infovis TUIs' performances. Of the literature surveyed, only two studies evaluated objective user performance comparatively against other forms of interaction. Two papers was excluded from this count: G-nome Surfer 2.0 was evaluated with the exclusion of the only tangible object [214] and Embodied lenses were evaluated on a simplified canonical query task [124]. Phylo-Genie was comparatively evaluated for phylogenetic learning against a traditional paper-based condition to determine correctness [204]. A controlled experiment was also used to assess time-on-task for four experimental conditions in a visual organisation task [108].

UE Evaluating user performance was the most common evaluation scenario adopted in the infovis TUI literature. The methods used include information evaluation, usability testing, laboratory questionnaires, observations and interviews. Usability tests have users perform a set of predetermined tasks while their performance is observed. The literature shows that usability testing is regularly combined with questionnaires to assess experience as well. G-nome Surfer 1.0 [211] usability was tested with a simplified visual bioinformatics task to determine that the tool supports smooth transition between the various levels of representation, the presentation of results and task completeness. The usability test was later followed with a questionnaire to register the user's options on the task and tool. A usability test was also applied to assess the TQI system's performance against a GUI-based query interface, which was followed by a post-task questionnaire [245]. EMERGE [232], a physical bar chart system, used usability testing followed by a questionnaire to explore the user interactions with the dynamic bar charts.

User interactions were also evaluated with a usability study in a query interface that utilised active tangible [253]. Venice Unfolding [169] and ArtVis [53] adopted similar techniques to evaluate user performance and experience. Questionnaires also frequently follow controlled experiments [108, 204], to asses subjective user experience along with objective user performance [108, 204].

Informal evaluations are typically based on domain experts' feedback performed in an informal setting, i.e. no set tasks and users are largely encouraged to try out the system. Vispol was informally evaluated by police officers, the domain experts for crisis management in a police station [147]. Practicing and academic urban planners (along with other visitors) experienced Urp in a demonstration or hands-on to provide feedback [250]. Sparse tangibles, a collaborative gene network exploration system, was informally evaluated by expert biologists who contributed comments about the system [9]. In one case, an informal user evaluation was carried out with various users, not experts, to collect initial feedback when using tangible lens on an interactive surface [134].

Laboratory observations were combined with questionnaires in ArtVis, where participants were asked to explore the artwork data set for a set time and to measure user experience [53]. Casual health information active tangibles were previously mentioned as they were evaluated for users' understanding of visual data and its analysis, and a second evaluation validated the results using field observations [40]. Field observations were also recorded as users in a museum exhibit interacted with tangible rings on an interactive surface and a digital touch version [149].

The majority of evaluation carried out in infovis TUI research are categorised under UE. Two papers were not included in the account since their systems did not include a tangible in their evaluation [210] or the evaluation of the system was carried out on a simplified task not representative of an infovis [124]. Controlled experiments for the purpose of objectively evaluating performance are rarely applied, and when they are it was largely to measure time-on-task and lacks explanation of quantitative or qualitative cause. VDAR evaluations were carried out by some of the studies to assess the system's support for the infovis task or the interaction with the task. CTV evaluations were rarely conducted, while VA evaluations were not noted in the literature. For collaborative interfaces, CDA evaluations were performed to understand how the interface supports collaboration.

2.6.4 Summary

Evaluation is a key component of HCI and is commonly performed at various point in a system's development lifecycle (formative and summative evaluations). Current infovis TUIs (see Table 2.12) were largely evaluated to assess UE with usability tests, informal evaluations, observations and questionnaires. Fewer studies evaluated UP, and, when applied, controlled experiments were the favoured approach. Interviews and observations were commonly used for UWP and VDAR evaluations.

This research carried out formative and summative evaluations throughout its lifecycle. Semi-structured interviews were carried out to explore the case study domain and elicit functional requirements (see Section 4.2), while an early controlled experiment identified strengths and weaknesses of touch versus mouse input for a case study task (see Section 4.3). Later, two common TUI technologies were systematically evaluated for suitability (see Sections 6.2 and Section 6.3). An initial prototype was informally evaluated next to elicit subjective feedback on usability and design (see Section 6.4.12). Once a final TUI was refined, users' performance was assessed using a comparative controlled experiment and augmented with laboratory observations and a questionnaire (see Chapter 7).

Chapter 3

Case Study: Quantitative Genetics

3.1 Introduction

This chapter provides an overview of quantitative genetics - a popular strategy adopted for examining the relationship between the genome and traits, particularly those related to diseased traits. The first section introduces the human genome and addresses terms that help to clarify the concept of quantitative genetics. The following section describes QTL studies, an analytical process for identifying those regions of DNA that affect traits that are associated with a larger area of the human genome. It pays particular attention to eQTL including tools adopted for its mapping and visualisation. A detailed look is given next of two tools developed to specifically address the issue of interpreting eQTL association mapping results by using non-traditional means of visualising said results. The chapter ends with a summary.

3.2 The Human Genome

Each human cell stores DNA in its nucleus, which is organised into chromosomes. DNA is a long sequence of molecular bases that contain the genetic information used for the synthesis of protein and thus facilitate biological functions. The molecular bases that make up the DNA are adenine (A), cytosine (C), guanine (G) and thymine (T). The two strands that form a DNA's double helix structure consist of the aforementioned sequence with its complementary strand running in the opposite direction where the adjacent bases bond together forming base pairs. Each base pair is formed from two complementary nucleotides - in DNA, that is A with T and C with G [194].

The genome functions at a subunit level, better known as a gene; the human genome is comprised of 20,000-25,000 genes of varying lengths. Each individual receives a maternal and a paternal genome copy, thus two 'copies' of each gene are maintained. The nucleus stores the genome for the lifetime of the cell, whereas the production of protein is a dynamic continuous process and therefore it is maintained outside the nucleus by cellular machinery known as ribosomes. The genetic information reaches the ribosome by transcribing the sequence into a temporary structure, ribonucleic acid (RNA), which carries it from the nucleus to the ribosome. The RNA sequence is then translated by the ribosome into a string of amino acids that form protein. The combined process of transcribing and translating genetic information is known as gene expression [194].

All the cells in the human body contain the same genome, however the information contained within the genes inform the cell about how to produce a protein that provides the structure and function for that cell, i.e. it informs the level of expression of each gene in the human genome. Gene expression is regulated by sequences within the DNA and a set of regulatory proteins. The regulation of a gene can affect the frequency and rapidity of transcription, as well as the RNA's ability to access a piece of DNA [194].

3.2.1 Phenotype and Genotype

The genetic information contained within the DNA (genotype) controls an organism's observable structure or function (phenotype or trait). Phenotypes are largely based on their underlying genotype, however the expression of those genes is also influenced by environmental factors. For example, a trait such as hair colour is encoded in one's genes but exposure to the sun can cause hair to change in colour. A trait can be physical, behavioural or have a predisposition to disease.

These traits are determined by alleles, a set of genetic information of the different forms of a gene [230].

For instance, the wrinkled or smooth shapes of a seed are the phenotypic trait variations of a gene's different forms; we can use w to describe the wrinkled seed and R to describe the round seed. Given the two alleles for the seed colour trait, we can have one of the following combinations from each of the inherited genes: RR, ww and Rw. A plant with two of the same allele, RR and ww, is known to be homozygous and will result in a round and wrinkled seed respectively. In the third combination, Rw, two different alleles will interact and in this case R will mask the w allele trait (resulting in a round seed) and this is known as heterozygous [230].

The previous example is that of a Mendelian trait which is the result of a variation in a single gene and are classified as having that trait or not. This is not the case for most phenotypic traits, such as height, where variations are shown along a continuous pattern of phenotype distribution and are known as quantitative traits. Quantitative traits result from a cumulative interaction of small variable effects of various genes and are referred to as QTL [194, 230].

3.2.2 Genetic Variations

The human genome has approximately three billion base pairs, of which no two individuals are the same, not even identical twins [24]; the difference between any two individuals is estimated to be 0.1% [251]. That difference is what leads to the various differing traits between individuals and predisposes them to more complex traits, i.e. quantitative traits [224]. The importance of genetic variations lies in their use in medical genetic analysis of an outbred species where mating occurs between distantly related individualse.g. humans.

Genetic variations can range from mutations, deletions, insertions or polymorphisms in the DNA. A single nucleotide polymorphism (SNP) is the prevalent form of variation and makes up 90% of the human genetic variations. A SNP is a genetic variation where a single nucleotide is replaced by another when sequences are compared by position [224]. SNPs are randomly distributed over the genome and can occur in or outside genes with differing effects.

3.3 Quantitative Genetics Analysis

The detection of QTL is a difficult feat compared to Mendelian loci (i.e. a variation in a single gene or locus) and is concerned with combining genetic variations and trait genotypes to investigate the individual genes that make up the QTL. There are various methods used for the detection of QTL, and the basic shared principle underlying these methods is the application of correlative statistics between partitioned classes of a population based on genotype at genetic variation loci to determine the degree of association between the classes in respect to the quantitative trait. A statistically significant association between the classes suggests potential QTL [233].

An eQTL underlines the genetic variations to multiple interacting genes as QTL does, however the mediated trait is due to the regulation of a gene's level of expression. In eQTL mapping, the abundance of transcripts in a particular cell is quantified and treated as a quantitative trait using conventional QTL methods. The resulting association between transcripts and genetic variations can then be correlated with quantitative traits. The types of gene expressions examined in eQTL studies is dependent on the approach undertaken for the study and the number of individuals processed. The number can range from the very little (e.g. Welsh et al.[260] analysed two types of tissues from normal and cancerous cells) to much more (e.g. Chen et al. [35] examined 46 genes).

Typically eQTL are mapped using a genome wide association studies (GWAS), where new loci are identified without previous knowledge of regulatory regions and covering the whole human genome [172]. GWAS examines DNA of individuals with varying phenotypes for a particular trait, or most commonly a disease. These studies aim is to identify risk loci within the genome that affect quantitative or Mendelian traits. The identification of the risk factors makes it possible to predict illness in individuals, underpin the genetic susceptibility and to develop preventative treatments (e.g. stratified medicine). GWAS has been used to explore the susceptibility locus for diseases such as epithelial ovarian cancer [260], asthma [167], autism [3] and prostate cancer [35].

Human eQTL studies have largely been performed on blood cells or cell lines (e.g. [144, 229]) to understand gene expression and cell regulation. However,

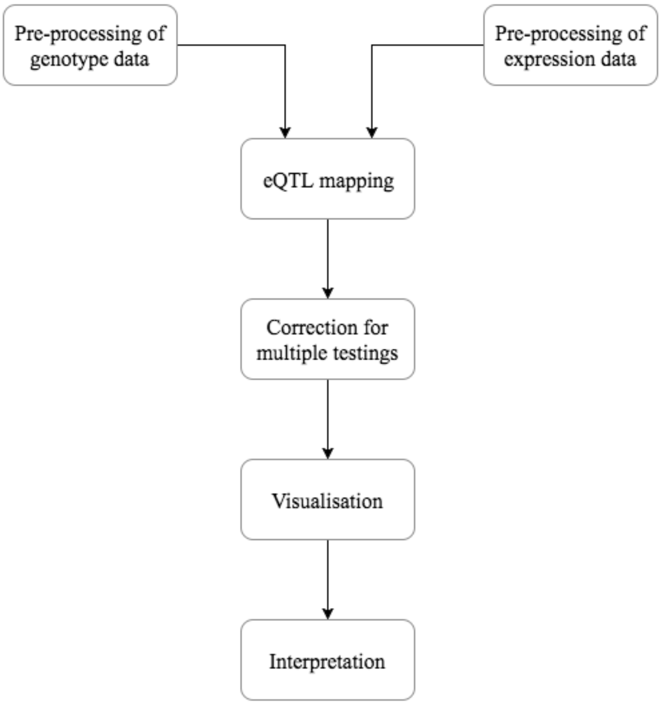

Pre-processing of genotype data
Pre-processing of expression data
eQTL mapping
Correction for multiple testings
Visualisation
Interpretation

Figure 3.1: Step-by-step protocol for eQTL analysis in humans reviewed by and adapted from Franke et al. [69].

gene expressions are specific to cell types and its regulatory control could also be dependent on cell type. Regulatory variations of gene expression inform the discovery of pathways that are causal for diseases [172]. Predisposition to disease is typically an outcome of gene networks affected by a number of genetic variants that are largely distributed across the human genome. The subsequent eQTL analysis of gene networks or sub-networks aids in the identification of regulatory eQTL that affect genes in this network (e.g [36]).

The general steps for eQTL analysis have been reviewed by Rockman et al. [196] and Franke et al. [69]. The step-by-step protocol provided by Franke et al. [69] for the analysis of genomic data (see Figure 3.1) uses gene expression measurements from a natural population of unrelated individuals and SNP markers to provide insight and guide further investigation into disease susceptibility. The first two pre-processing steps ensure the conversion of genotype and expression data into appropriate formats to avoid false-positives, however, they are of little relevance in this research, as the data used is ready for mapping.

Statistical Package	Command Line	Graphical	Web-based
MatrixeQTL*†	PLINK	FastMap	SNPster*
R/qtl	Merlin	gPLINK	
snpMatrix	Qxpak.5*†		
eMap	associationGG*		

Table 3.1: eQTL analysis tools ordered by processing speed from fastest to slowest. (†) Tools developed specifically for eQTL analysis. ($\star$) Tools that could not have their speed measured [265].

Once the data has been pre-processed, eQTL association mapping is performed by analysing gene expression transcripts for association against SNPs. This step requires specification of the regulation type and correlation measures to be used. The eQTL mapping process necessitates multiple test correction, as tens of thousands of transcripts are associated with hundreds of thousands of SNPs. A false discovery rate (FDR) correction is recommended due to its effective control over false-positives in eQTL analysis.

Wright et al. [265] surveyed a number of software tools used for eQTL association mapping that are either specifically designed for that form of analysis or adapted for it. The tools range from statistical packages to stand-alone or web-based software [69, 265]. Table 3.1 arranges various mapping tools based on their interface type and orders them by processing speed from fastest to slowest.

The results of an eQTL study is a series of genetic variants and their gene expression associated statistical significance. These results can be visualised to prompt interpretation that may lead to further investigation. There are various tools (stand-alone or web-based) that are used for such a purpose, some of which provide plotting capabilities to visualise eQTL. For instance, PLINK [190], a separate Java-based graphical user interface, is integrated with Haploview [14] to visualise eQTL analysis results.

Figure 3.2 shows the association results of an eQTL study of the CNTN2 gene as expressed in the brain analysed using PLINK [190] and visualised as a Manhattan plot in Haploview [14]. The data points represent SNPs plotted against their chromosomal position and $-log_{10}\ p$ significance. A Manhattan plot is

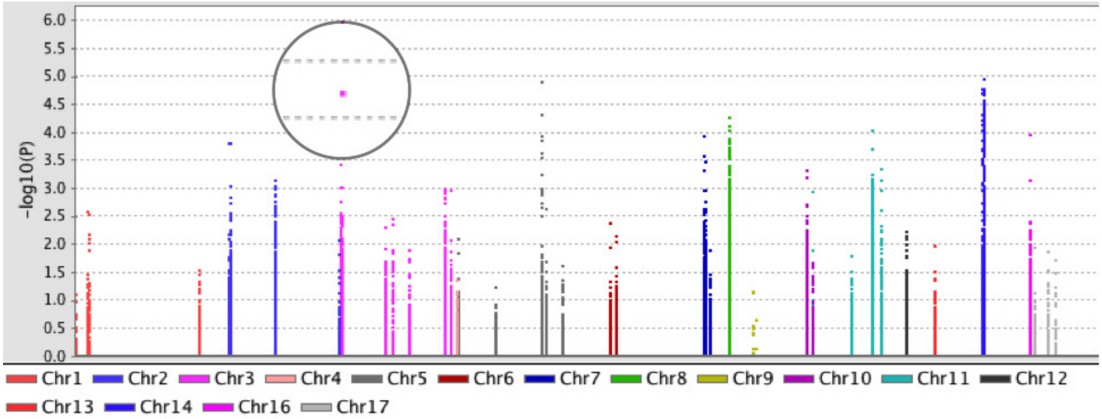


Figure 3.2: A Manhattan plot of the analysis results of our eQTL study of the CNTN2 gene as expressed in the brain. The data points on the plot represent SNPs plotted against their chromosomal position and $-log_{10}\ p$ significance. The data point at approximately $-log_{10}$ 4.7 is magnified with a lens to highlight its potential importance.

a logarithmically scaled scatter plot that is designed to highlight small variations from a normal range; variations of higher significance have higher logarithms and will be easily recognisable. For instance, the magnified data point in Figure 3.2 has approximately $-log_{10}$ 4.7 significance, this can potentially mean that this specific genetic variant is casual for variations in cell-cell adhesion molecules for the CNTN2 gene as it is expressed in the brain.

3.3.1 Quantitative Genetics at the University of Leeds

Research in quantitative genetics is carried out at the University of Leeds in various departments. At the Cancer Genetic Group, quantitative genetic is used to understand the genetic basis of cancers for the purpose of developing preventive strategies and treatments. In their work they aim to identify and asses how the human genome and the environment adversely affect gene regulation. The Population Biology and Control of Parasitic Diseases uses quantitative genetics to decipher genetic factors of the hosts of parasitic diseases that can potentially identify controlling genetic variants, i.e. resistant loci. For this type of human- or population-based researches, large field studies are typically undertaken to collect data under strict governmental and health regulations and shared by research centres.

The Centre of Plant Sciences hosts research on plant epigenetics involving the discovery of gene expression states that go beyond DNA sequencing differences. These states lead to variations such as cell types differentiation and phenotypic diversity. The identification of non-Mendelian epigenetic QTL in plant species can support and enhance agricultural planning and production. This is possible by repressing epigenetic QTLs when certain phenotypes are not needed and reactivating when conditions are suitable. Quantitative genetics is also used in the centre for the purpose of plant breeding. This process involves the selection of parents, breeding variability, selection of desired phenotypes and the synthesis of cultivars from the selection using quantitative indices and prediction equations.

Human, animal, and plant DNA are similarly coded with the four molecular bases, however there difference lies in how the bases are arranged, the number of chromosomes and polyploid. Genome size also varies, for instance the human genome consists of 3,235 mega-basepairs whereby castor beans consist of 320 mega-basepairs. These differences present various challenges to quantitative genetics, particularly due to recent advances in DNA sequencing. Research in QTL is relatively new at the university with more interest in the human genome, cancer genomics and diseases. The challenges of this work, such as the scale of the data set, compared to plant genomics aligns itself with the interest of this research. While access to human genome data might have proven difficult due to ethical restrictions, this was overcome with the announcement of the Biovis 2012 contest [103].

3.3.2 BioVis 2012 Data Set

The biological domain of Biovis 2012 contest [103] is eQTL association where a collection of genotype and gene expression data is provided for the identification of genetic variants of regulatory significance. This provided an opportunity to work with real data that has been spiked for the identification of causal eQTL. The data sets provided are as follows:

- A PED file which contains genotypic information on 500 unrelated individuals. Each row identifies an individual using a number of columns: individual ID, family ID, affection status, and the genotype information for each SNP included in the MAP file.
- A MAP file describes the 230,912 SNPs genotyped in the PED file and contains chromosomal locations, SNP identifiers (registered serial number, RSID, or a combination of their chromosomal location and position), and chromosomal positions.
- A PHEN file specifies alternate phenotypes for association mapping to replace the affection state column in the PED file. The phenotypes contained in this file are a quantitative measure of the gene expression of 44 genes as expressed in the blood and the brain for all of the 500 individuals listed in the PED file.

Using PLINK [190], an open source whole genome association toolset, eQTL analysis was achieved using standard linear regression in a population of unrelated individuals, which uses one SNP and one gene expression at a time. This was provided by the contest, where each file contained 230,912 rows - one for each SNP identified in the PED file. Forty-four genes were also chosen by the contest as they are highly expressed in the brain and represent genes that are relevant to psychiatric diseases. The genes were broadly categorised in three broad groups (see Table 3.2). The expression of these genes in the blood and brain were provided, resulting in a total of 88 gene expressions.

The eQTL mapping process, which analyses a large number of gene expressions for association against an even larger number of SNPs, necessitates correction for multiple tests that was achieved using PLINK's adjust command. The resulting file contained the unadjusted p-values as well as possible adjustments that included genomic-control, Bonferroni, Holm, FDR control, and Sidak adjustments for blood and brain expressions of 44 genes.

The result of PLINK's eQTL analysis is an association file for each gene expression. Each file associates a gene expression against genetic variants; each variant is given a calculated statistical value to indicate its significance to the gene expression. Tables 3.3 and 3.4 display the number of significant SNPs where

$p < 0.05$ for the 44 genes as they are expressed in the blood and the brain respectively. Typically, the result of the analysis process is pipelined to Haploview [14]. In Haploview, each result file is tabulated and can also be viewed in a Manhattan plot given a set of variables. Figure 3.2 shows a Manhattan plot displaying the association results of the CNTN2 gene as expressed in the brain using PLINK and Haploview.

3.3.3 Sources of Information

The Biovis contest premise was to visualise eQTL association mapping results in a manner that is helpful to researchers by utilising relevant information retrieved from external databases. PharmGKB and dbSNP were both recommended by the Biovis contest providers.

PharmGKB is a pharmacogenomic data resource for investigating genetic variations and their effect on drug responses, and it also provides interrelated data about genes, drugs, and diseases [162]. The data sets provided by PharmGKB include variant and clinical annotations. Variant annotations are curated manually from pharmacogenetic literature to associate genetic variants with drug responses. Each variant annotation is given a PharmGKB annotation ID, which uniquely

Cell-cell adhesion molecules	Neurotransmitter systems	
	Serotonin	Dopamine
AGRN, CNTN1, CNTN2, CNTN3, CNTN4, CNTN5, CNTN6, CNTNAP1, CNTNAP2, CNTNAP3, CNTNAP3B, CNTNAP4, CNTNAP5, NLGN1, NLGN2, NRG1, NRG2, NRG3, NRXN1, NRXN2, NRXN3	HTR1A, HTR1B, HTR1D, HTR1E, HTR1F, HTR2A, HTR2B, HTR3A, HTR3B, HTR3C, HTR3D, HTR3E, HTR4, HTR5A, HTR6, HTR7, SLC6A4	DRD1, DRD2, DRD3, DRD4, DRD5, SLC6A3

Table 3.2: The 44 genes chosen by the Biovis 2012 contest [103].

Genes	Bon.	FDR	Holm	Sidak	Genes	Bon.	FDR	Holm	Sidak
AGRN	12	25	12	12	HTR1F	27	29	27	27
CNTN1	1	1	1	1	HTR2A	3	1	3	3
CNTN2					HTR2B	23	23	23	23
CNTN3	11	11	11	10	HTR3A				
CNTN4					HTR3B	2	1	2	2
CNTN5					HTR3C	10	10	10	10
CNTN6					HTR3D	1	1	1	1
CNTNAP1	9	9	9	9	HTR3E	1	1	1	1
CNTNAP2					HTR4				
CNTNAP3	28	28	28	28	HTR5A	1	0	1	1
CNTANP3B					HTR6	3	1	3	3
CNTNAP4	8	8	8	8	HTR7				
CNTNAP5	7	6	7	7	NLGN1				
DRD1	1	1	1	1	NLGN2				
DRD2	2	1	2	2	NRG1	3	3	3	3
DRD3					NRG2				
DRD4					NRG3	2	2	2	2
DRD5					NRXN1	4	4	4	4
HTR1A					NRXN2				
HTR1B					NRXN3				
HTR1D	10	10	10	10	SLC6A3	2	2	2	2
HTR1E	1	0	1	1	SLC6A4	2	2	2	2

Table 3.3: The results of the eQTL study showing the number of possibly causal SNPs for each of the 44 genes as expressed in the blood. Various corrections (Bonferroni, FDR, Holm and Sidak) are also shown.

Genes	Bon.	FDR	Holm	Sidak	Genes	Bon.	FDR	Holm	Sidak
AGRN	38	47	38	38	HTR1F				
CNTN1					HTR2A				
CNTN2	8	7	7	8	HTR2B				
CNTN3	11	11	11	10	HTR3A				
CNTN4					HTR3B				
CNTN5					HTR3C	2	0	2	2
CNTN6	1	0	1	1	HTR3D	1	1	1	1
CNTNAP1					HTR3E				
CNTNAP2	1	1	1	1	HTR4	3	3	3	3
CNTNAP3					HTR5A	1	1	1	1
CNTANP3B					HTR6				
CNTNAP4					HTR7				
CNTNAP5					NLGN1	1	1	1	1
DRD1	2	1	2	2	NLGN2				
DRD2					NRG1	3	3	3	3
DRD3					NRG2				
DRD4					NRG3	5	5	5	6
DRD5	8	8	8	8	NRXN1	6	5	6	6
HTR1A					NRXN2				
HTR1B					NRXN3	29	31	29	29
HTR1D					SLC6A3	2	2	2	2
HTR1E	4	2	4	3	SLC6A4				

Table 3.4: The results of the eQTL study showing the number of possibly causal SNPs for each of the 44 genes as expressed in the brain. Various corrections (Bonferroni, FDR, Holm and Sidak) are also shown.

associates a drug with a SNP, identified by its RSID and/or gene location. This association is further linked with the curated literature, association significance, and study parameters. The study parameters are in turn uniquely identified and include: study type, size, allele frequency, p-value, and ratio statistics.

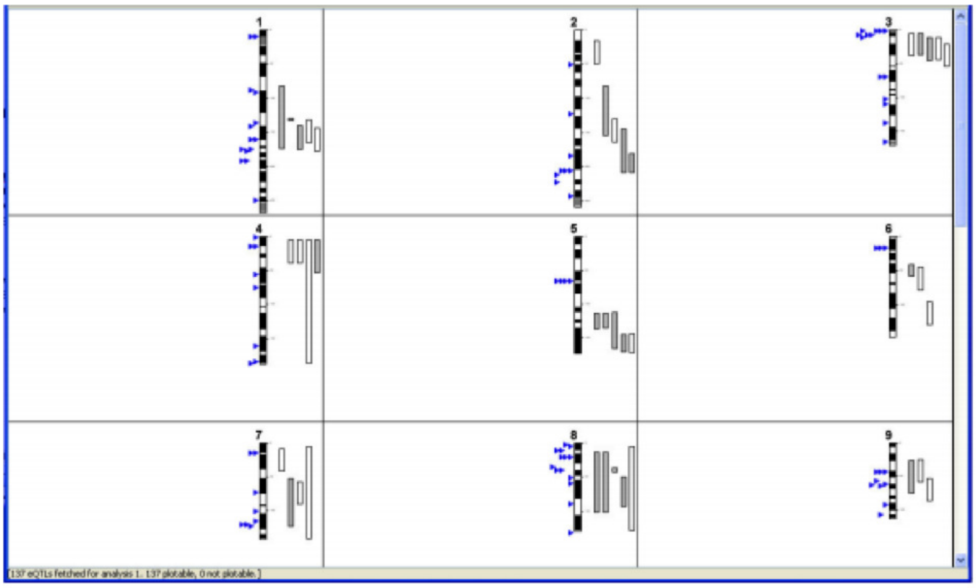
The SNP database (dbSNP) is a large database of simple genetic polymorphisms of various organisms maintained by the National Centre for Biotechnology Information (NCBI) [217]. In its latest build (137) there are 59,060,743 simple variants in the human genome, of which 85.7% were SNPs. It is possible to download dbSNP and use it on a local machine, however due to the sheer size of the database (500GB) and the limited number of relevant SNPs, entrez programming utilities (eUtils) were used to search and retrieve SNP data. eUtils are a set of services that provide an interface to the Entrez databases, including dbSNP.

3.4 Visualisation Tools for eQTL

As it was mentioned previously, the results of an eQTL study are typically explored through a Manhattan plot or a table that summarises said results. This, in turn, requires the examination of one quantitative trait at a time, thus providing limited capabilities to compare and contrast given the large number of genes examined in eQTL studies. This section discusses two tools (eQTL explorer [168] and eQTL viewer [272]) that have been developed to explore the association given those constraints. The tools purposely aid the process of visualising and interpreting the study results given a traditional setup.

3.4.1 eQTL Explorer

eQTL explorer is a standalone Java based tool that visually integrates the results of genome-wide expression and physiological QTL mapping to assist the generation of biological hypotheses [168]. Once the results are pipelined to the visualisation tool, the eQTL are displayed based on their genomic physical locations along the chromosome, while also displaying known physiological QTL (pQTL). This overview allows the user to view the association along the chromosomes that can be browsed through. The eQTL explorer's interface has a genome panel (see



[137 eQTLs fetched for analysis 1. 137 plotable, 0 not plotable.]

Figure 3.3: eQTL Explorer's genome panel provides an overview of the example data set showing eQTL and pQTL along 9 chromosomes. eQTL are displayed as coloured arrows ordered by p-value, while pQTL are represented as vertical bars.

Figure 3.3), where data is retrieved from a database that fetches the data set based on certain criteria selected by the user, such as specific tissue samples, p-values or eQTL types.

A detailed view of a chromosome is presented in a chromosome panel that displays a single chromosome with access to eQTL and pQTL annotations. QTL annotations are displayed by hovering the mouse cursor over a particular QTL; external resources can also be accessed by clicking on a QTL and selecting any of the options displayed. An option to display the data in a table format is also provided, as this is the typical view adopted by analysts. Both visualisation options, chromosome and table, provide export options and links to external databases.

The overviews provided by eQTL explorer, genome and chromosome panels, allow the interpretation and development of a hypothesis that signifies the relationship between eQTL and pQTL. The advanced options provided allow the customisation of the displayed results, such as comparing the results of a number of experiments. Nevertheless, the visualisations adopted may prove cumbersome as the scale of the data set increases.

3.4.2 eQTL Viewer

eQTL viewer is a web-based visualisation tool that addresses the need to identify the relationship between statistically significant regions of genetic variations and transcripts [272]. The visualisation places emphasis on gene-gene relationships. Using scalable vector graphics (SVG), the mapping results are pipelined from a mapping tool and visualised as a scalable and annotated two-dimensional plot; the vertical and horizontal axes correspond to the genomic location of transcripts and candidate genes that include eQTL respectively. An eQTL is shown as a small black bar that is superimposed with either a green dot to denote an eQTL that is part of a transcriptional factor for regulating networks, or a red dot to indicate an eQTL that can form a protein complex with the vertically aligned transcription (see Figure 3.4).

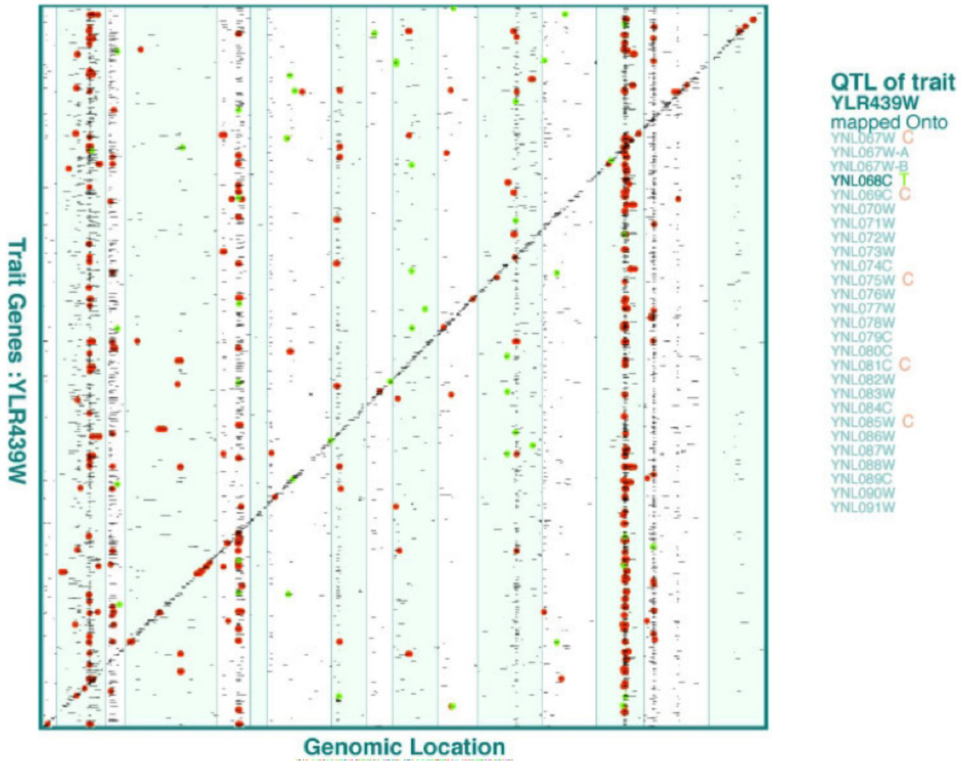


Figure 3.4: Using the yeast data set example, Zou et al. [272] displays eQTL information using eQTL Viewer.

The plot's interactivity allows its users to zoom in to study a region of gene-gene interaction or a single genetic variation of significance. In contrast, zooming out provides an overall overview of the experimental data set. Pointing at an eQTL would display the genes involved in that particular interaction, i.e. the transcription and candidate gene. These in turn may be linked to their annotations on public databases specified in scripts written by the tool's users. A search function is also provided to query transcript names that are highlighted on the plot if found. The interaction techniques used for navigating the plot involve manipulating a mouse and keys on a keyboard to zoom, pan, search and select regions on the plot.

The general overview of an analysed data set provides emphasises on gene-gene interactions for samples collected from the same cell, which is not always the case for complex trait analysis where samples collected from different tissues are compared and contrasted to further understand the genetic and clinical basis of eQTL.

3.5 Summary

QTL analysis examines the relationships between the genome and complex traits and identifies these locations in the genome. Similarly, eQTL analysis identifies locations in the genome that affect the regulation of a gene's expression. This chapter provided an introduction to human genomics to help clarify the concepts of QTL and eQTL (see Section 3.2). It also examined the general steps for eQTL association mapping, as well as the tools used for analysis and visualisation (see Section 3.3). The data set used in this research is based on those provided by the Biovis 2012 contest [103], which were analysed using PLINK [190] and inspected for matches (see Section 3.3.2). The contest's data sets were utilised in this thesis to fabricate test data for an experiment (see Section 7.2.2). Analysis results are typically viewed in a Manhattan plot, however other tools had visualised results with pQTL (eQTL explorer [168]) and transcripts (eQTL viewer [272]). These tools were used to explore varying visualisation representations with quantitative genetics analysts (see Section 4.2). The visualisation of eQTL results was used in this research as a case study for infovis.

Chapter 4

Interface Requirements

4.1 Introduction

This chapter describes the methods undertaken to elicit eQTL analysis requirements and explores its application in a graphical user interface (GUI) setting. The first section details the methods adopted to gain a better understanding of the QTL/eQTL analysis and interpretation process. It also identifies high-level tasks and themes for quantitative genetics and eQTL. These are then used along with the analysis scenario to elicit the functional requirements for an eQTL application with a focus on data visualisation and interpretation. The next section outlines the development of a GUI to explore novel functionalities and interactions. This is followed with a comparative evaluation that assess one of the eQTL tasks (combine files) using mouse and touch inputs. The chapter ends with a summary.

4.2 Interviews

The results of an eQTL analysis include a list of eQTL files that associate gene expressions with genetic variants based on their genetic significance. These results can be visualised to obtain insights that may lead to further investigations. The results are typically displayed as a Manhattan plot or summarised in a table. A Manhattan plot is a logarithmically scaled scatter plot that is designed to highlight

Interview	Analyst	Expertise
1	A1	Plant breeding and genetics
2	A2	Statistical genetics methodology for epidemiology studies
3	A3	Statistical genetics methodology for the discovery of genes involved in the aetiology of complex diseases
4	A2 and A4	Statistical genetics methodology for epidemiology studies

Table 4.1: List of interviews conducted, along with the analysts and their expertise.

small variations from a normal range; variations of higher significance are also easily recognisable.

Qualitative interviews are conducted with quantitative genetics analysts to gain a better understanding of the analysis and interpretation process. A semi-structured approach is undertaken for the interviews to gain a broader understanding of the analysis process. This approach is chosen over others, such as questionnaires and focus groups, to provide room for adjustment according to the interviewees' responses and the pursuit of matters of interest.

4.2.1 Method

Initially, three analysts were interviewed. The first analyst mapped QTL for plant species, and the other two had expertise in human epidemiology studies and were progressing towards adopting eQTL in their work. All the interviews were held at the analysts' place of work and audio recorded. The interviews lasted from 40 to 90 minutes. With the second analyst, a follow-up interview lasting 20 minutes was also conducted. This was to discuss specific eQTL tools (eQTL explorer [168] and eQTL viewer [272]). Another analyst also took part in that session. See Table 4.1.

The interviews were structured around the following themes:

1. Introductions
2. Purpose of the QTL anlysis process
3. Analysis prerequisites, such as data set type, origin, tools and practices

4. Analysis types and tools
5. Analysis results
6. Computational setup
7. Suitability of the Biovis 2012 contest's data set [103] as representative data

Since the first interview was conducted to gain a general understanding of quantitative genetics, the other interviews proved more important in terms of identifying requirements relevant to eQTL.

4.2.2 Implications

The interviews were transcribed and analysed to identify high-level tasks and themes for quantitative genetics in general and eQTL in particular. The following sections report on the interviews and discusses their implications.

4.2.2.1 Scale

Within an eQTL study, tens of thousands of gene expressions are typically associated with hundreds of thousands of genetic variants. A2 indicated that unlike QTL and GWAS, eQTL works with a much larger number of outcomes. The analysts further explained the effect this has on the research, specifically how eQTL that are acting from a different gene are avoided owing to the increased number of genes that are investigated and the growing net of interactions.

> **A2**: The difference with eQTL is, instead of looking at one outcome, which is our disease state, we're looking at hundreds or thousands of outcomes because we're looking at the genes that transmit across the region. But, obviously, eventually people are going to want to look at that but I think if you do that in a completely hypothesis-free manner you're going to have such a problem of multiple testing that people are steering clear of it.

A3 also remarked on the scale of the data sets as well as the process of viewing one gene expression file at a time on a Manhattan plot, stating that the process is time consuming and complicates the interpretation process. A2 and A3 reported that issues with scalability are limited to not only the analysis process but also the results that are produced, which are dependent on the adopted multiple correction techniques and the neighbouring functional genetic variants. While traditional tools (e.g. MS Excel or Haploview [14]) work well when exploring a limited number of results, individually viewing gene expression files can prove to be time consuming and complicates the process of results interpretation.

4.2.2.2 Comparison

When A2-4 were asked about the purpose of retrieving gene expressions from different tissues, as is noted in the Biovis [103] data set where gene expressions were collected from blood and brain tissue, the analysts' unanimous response acknowledged the importance of correlation and its influence. A2 responded as follows to a question probing the significance of using different tissue samples.

> **A2**: Well, we use different tissues as we don't know ... It is still kind of emerging, really, to what extent important variants influence gene expression everywhere, or is it just tissue-specific? So if we're looking at a specific disease like melanoma, we might be interested in looking at melanocytes, ordinary skin tissue, but also you might look at cells that have to do with immunity.

A3 made a similar remark when talking about multiple gene expressions collected from the same tissue but at varying stages of cancer progression.

> **A3**: We show a Manhattan plot for melanoma and then we can split tumour melanoma into different groups, some are at a more advanced stage and some are not and then that Manhattan plot analyses one for each group. Another stage in the gene expression analysis is to look at them in a combined way; how they co-express each other.

At the follow-up interview, A4 indicated the process of comparing and contrasting eQTL study results for their dependencies on a disease.

> **A4**: There are different things that you might look at, depending on the disease. There are now, more papers coming out that show . . . There was one, in the last couple of months that show that quite a lot of the associations might be cell-specific. So, you might have variants that actually are quite predictive of gene expression in certain cell types but not necessarily in others.

The analysts' responses indicate the importance of the evidence resulting from comparing study results of gene expressions collected from different tissues. Comparisons can be conducted between tissue samples of the same gene, different genes or disease-specific genes.

4.2.2.3 Heterogeneity

During analysis, heterogeneous information may be accessed to gain prior knowledge, confirm findings or provide an explanation for the association. The analysts all recognised the importance of accessing readily available information. A3 recalled an instance where access to external information may be helpful.

> **A3**: It is biological relationships you are looking at - the relationship between a SNP and gene expression - and then you correlate, you can speculate on causality. Is it causal or not causal? That's something difficult to say from the statistics actually. When you find a statically significant relationship you can not say it is causal. You need more evidence, like functional assays.

A3 added the following in response to the question about the importance of providing access to external resources.

> **A3**: I think what you need is something to back up your findings, to be able to explain it in a concise way, to give justification to each word you use.

From the analysts' responses, it can be concluded that access to external heterogeneous data can provide a means to reference or confirm eQTL outcomes.

4.2.2.4 Sharing

Biological research results are often shared on public or private databases. According to the interviews, this is also the case for QTL and eQTL studies. A2 stated that in their current research, they are reliant on publicly available data.

> **A2**: At the moment, we're purely relying on publicly available data but we may, down the line, end up doing some of our own. The publicly available data is crucial, really, to this [groups' research].

In the follow-up interview, A4 explained the importance of information sharing in their research.

> **A4**: Why should the MRC fund us to do an eQTL experiment on a certain melanocyte, say, on 1000 people, and then we keep the data to ourselves and then they fund somebody else in London to do the same thing because they're interested in similar things?

Despite the clear importance of sharing eQTL study results, A2 reported that they are not familiar with a systematic resource for sharing these results.

> **A2**: At the moment, it seems a little bit ad hoc, as far as I know, but I can imagine somebody might well, before long . . . I've not come across anywhere that's a nice central resource that tells you where to go but that would be very useful.

A3 also acknowledged the importance of sharing and its commonality in the field of genetics. This highlights the importance of sharing eQTL study results.

4.2.2.5 Visualisation

Two visualisation tools for eQTL (eQTL explorer [168] and eQTL viewer [272]) were presented to A2 and A4 to understand whether visualisations other than a typical Manhattan plot are used or may be of potential use when interpreting analysis results. A4 explained that these tools may be useful for biologist, but they are not particularly suited for analysts.

> **A4**: See, in our case, something like this wouldn't be particularly useful. This is more like a biologist thing, where they want to kind of look at the picture and play around. What we need is to come to some kind of conclusion and quantify what's going on, rather than saying, 'Oh. This looks interesting and this looks interesting.' We need to be able to say, 'This is significant.'

A2 acknowledged the importance of plotting in response to the question of the role of visualisation in eQTL analysis and interpretations.

> **A2**: But, nonetheless, the plots are useful to see. So, one of the things we use . . . When we do the whole genome study, you get the occasional SNP that's significant all on its own and usually when you look at it, it's either very rare, or for some reason, it shouldn't have gotten through quality control. There's something wrong with it. So, these plots are very useful at actually being able to see the pattern and this is clearly a genuine approach. So, we can immediately see. Yes. So, a Manhattan plot is a useful visual aid for what we want to do.

A2 also emphasised that being able to combine and contrast eQTL results would simplify the analysis and interpretation process.

> **A2**: For eQTL, I think it be really similar except that I don't know how you do this but the thing that is different is you have a plot like this essentially for every transcript and we'll see the transcripts correlated as well. You know so they might be in the same gene or . . . So, I suppose we could easily plot something like this from the p-value results we get from our programs like PLINK but that would just give us a one transcript at a time. So, I suppose any kind of clever way of somehow combining or comparing those would be helpful.

This confirms the importance of visualisation for viewing results and how they could help analysts identify patterns when combining and comparing expressions.

4.2.3 Analysis Scenario

An eQTL study involves the collection and analysis of hundreds of thousands of genetic variants, which regulate how a gene would be expressed in a trait, from a number of individuals as well as gene expression measurements. The mapping process runs a single gene expression measurement against genome-wide genetic variants, resulting in significance values associating each genetic variant with the gene expression. The key aims of eQTL analysis include identifying risk genes for diseases so that appropriate treatments may be chosen for specific patients, i.e. stratified medicine, as well as identifying gene regulatory networks.

For eQTL data visualisation, biological analysts typically adhere to the following procedure. First, they open files from eQTL bioinformatics calculations to display the output for each gene expression in a table and/or Manhattan plot. The patterns are then interactively investigated by scrolling/panning/zooming the tables and/or Manhattan plots. Multiple files can be simultaneously viewed in different windows. In one of the open windows, the data is filtered (e.g. a specific chromosome is selected), thresholds are adjusted (e.g. the significance threshold) or the analyst drills down for additional information stored in external data sources (e.g. diseases known to be associated with specific genes or genetic variants). The files are explored independently, and patterns shared with other gene expressions are only discerned after individually exploring each file. Therefore, a clear improvement enables combining multiple files so that similarities can be identified in order to determine genetic variants that are significant across a set of gene expressions.

4.2.4 Functional Requirements

The above procedure highlighted the need for an interface that interactively visualises eQTL analysis results. The requirements outlined here focus on eQTL analysis where data visualisation plays an important role. The details are, of course, specific to eQTL. However, the tasks that users perform are generalised to those performed in other visualisation applications. This is shown by the similarities between the eQTL tasks and the tasks identified in Brehmer and Munzner's wide-ranging review [22] (see Table 4.2).

Generic Task	eQTL Task
Encode	Open/close files
Change	Switch windows
Navigate	Scroll, pan and zoom
Select	Select genetic variants
Import	Access external sources
Arrange	Organise windows
Filter/Change	Filter data
Aggregate	Combine files
Derive	Match significance across files

Table 4.2: Generic visualisation tasks [22] and specific tasks for eQTL infovis.

Users should be able to perform the following functionalities in the eQTL data visualisation application:

1. Open and close gene expression files.
2. Choose whether to display the file in plot or table windows.
3. Navigate plots and table to detect patterns.
4. Select genetic variants (SNPs) as rows in a table or data points in a plot.
5. Drill down to external data sources for selected genetic variants.
6. Organise windows of various files on the display.
7. Filter data sets.
8. Combine gene expression files.
9. Determine the significance of genetic variants across gene expression files.

4.2.5 Data Requirements

To provide the above functionalities, the application must have access to eQTL study results typically provided by various analysis tools. For the purpose of this research, we only consider results retrieved from PLINK [190]. This tool was previously used in this research to analyse the Biovis 2012 contest's data sets [103] (see Section 3.3.2). Each resultant gene expression file provides information about the analysed genetic variant (RSID, or a combination of their chromosomal location and position), chromosomal location, position and a calculated statistical value to indicate the variant's significance to the gene expression.

4.3 Graphical User Interface

A GUI is developed to experimentally explore novel functionalities and interactions. The initial iteration for developing the GUI involves a wireframe mockup to illustrate initial design ideas. The main goal of this iteration is to re-establish key interface functionalities. The desktop application is then developed in Java using JFreeChart [74], a free Java class library for generating various chart types. The remainder of the sections describes how the functional requirements are implemented for the GUI.

4.3.1 Open/Close Files

PLINK analysis results are usually viewed by opening a single file from a file explorer along with a MAP file of the genetic variants. For the purpose of speed, the experimental data sets provided by Biovis 2012 [103] are analysed in PLINK and formatted to automatically upload into the application (see Section 3.3.2).

4.3.2 Switch Windows

Study results are typically viewed in a table or visualised in a Manhattan plot, with users switching between windows as the analysis progresses. In the GUI, once the files are loaded into the application, the user is presented with a grid of Manhattan plot thumbnails (see Figure 4.1). Each thumbnail represents a gene expression file which can be opened by clicking on it. Thumbnails make it easier and faster for the user to look at a file and determine possible combinations without having to open the file. Once a gene expression file is opened, a Manhattan plot is displayed by default with the option to switch to table view using tabs (see Figure 4.2).

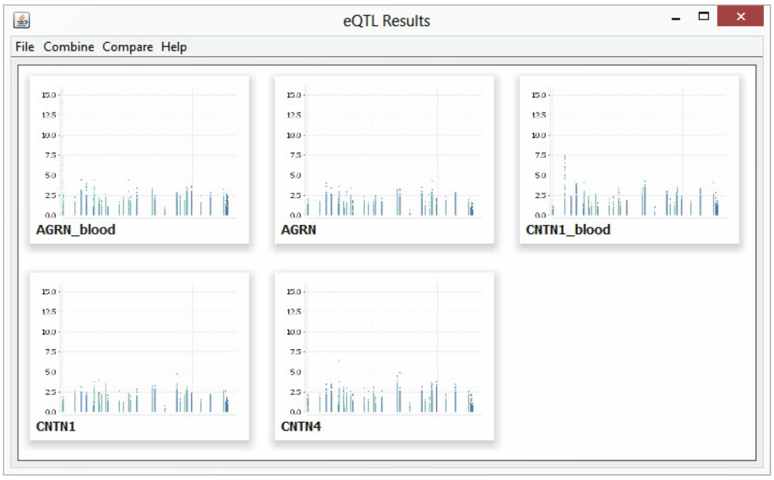


Figure 4.1: A grid view of the uploaded gene expression files.

4.3.3 Scroll, Pan and Zoom

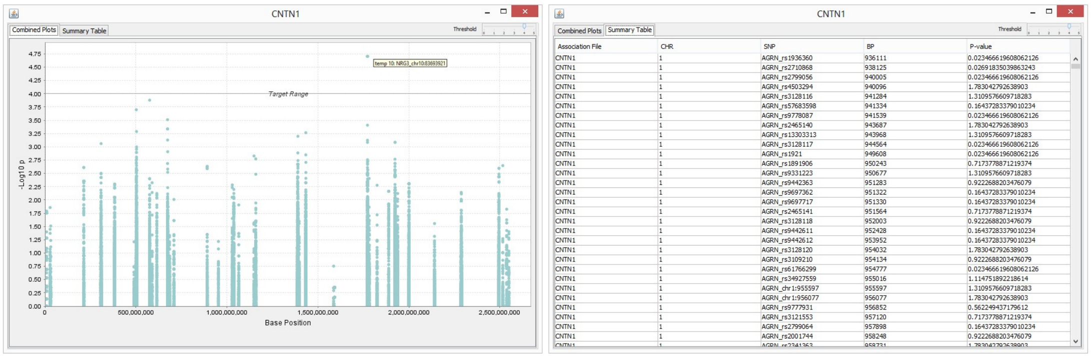
Plots and tables are navigated by panning and zooming to alter the users' viewpoint and increase their understanding. In the application, plots are zoomed either via a pop-up menu or a mouse drag on the displayed view. They are also panned by holding down a modifier key (CTRL on most platforms and ALT on Mac OS) and dragging the mouse. Tables are navigated using vertical scrollbars, where the thumb is dragged. Small amounts of scrolling are also achieved using the arrow key buttons.

4.3.4 Select Genetic Variants

In the plot window, additional information about the genetic variant that is not readily available is accessed by hovering the mouse cursor over a data point to display its infotip (see Figure 4.2). Rows are highlighted by selecting a genetic variant via a mouse click.

4.3.5 Access External Sources

External data sources are accessed when exploring eQTL to better understand the significance of a gene or genetic variant. External information is retrieved from the table view by selecting a row and clicking on a hyperlink, where an



Association File	CHR	SNP	BP	P-value
CNTN1	1	AGRN_rs1936360	936111	0.023466619608062126
CNTN1	1	AGRN_rs2710868	938125	0.02691835039863243
CNTN1	1	AGRN_rs2799056	940005	0.023466619608062126
CNTN1	1	AGRN_rs4503294	940096	1.783042792638903
CNTN1	1	AGRN_rs3128116	941284	1.3109576609718283
CNTN1	1	AGRN_rs57683598	941334	0.16437283379010234
CNTN1	1	AGRN_rs9778087	941539	0.023466619608062126
CNTN1	1	AGRN_rs2465140	943687	1.783042792638903
CNTN1	1	AGRN_rs13303313	943968	1.3109576609718283
CNTN1	1	AGRN_rs3128117	944564	0.023466619608062126
CNTN1	1	AGRN_rs1921	949608	0.023466619608062126
CNTN1	1	AGRN_rs1891906	950243	0.7173778871219374
CNTN1	1	AGRN_rs9331223	950677	1.3109576609718283
CNTN1	1	AGRN_rs9442363	951283	0.9222688203476079
CNTN1	1	AGRN_rs9697362	951322	0.16437283379010234
CNTN1	1	AGRN_rs9697717	951330	0.16437283379010234
CNTN1	1	AGRN_rs2465141	951564	0.7173778871219374
CNTN1	1	AGRN_rs3128118	952003	0.9222688203476079
CNTN1	1	AGRN_rs9442611	952428	0.16437283379010234
CNTN1	1	AGRN_rs9442612	953952	0.16437283379010234
CNTN1	1	AGRN_rs3128120	954032	1.783042792638903
CNTN1	1	AGRN_rs3109210	954134	0.9222688203476079
CNTN1	1	AGRN_rs61766299	954777	0.023466619608062126
CNTN1	1	AGRN_rs34927559	955016	1.114751892218614
CNTN1	1	AGRN_chr1:955597	955597	1.3109576609718283
CNTN1	1	AGRN_chr1:956077	956077	1.783042792638903
CNTN1	1	AGRN_rs9777931	956852	0.562249437179612
CNTN1	1	AGRN_rs3121553	957120	0.7173778871219374
CNTN1	1	AGRN_rs2799064	957898	0.16437283379010234
CNTN1	1	AGRN_rs2001744	958248	0.9222688203476079
CNTN1	1	AGRN_rs2341363	958731	1.783042792638903

Figure 4.2: A display window with the options to switch between a Manhattan plot (left) and summary table (right). The cursor is hovered over a data point in the plot to display its infotip.

external resource (e.g. dbSNP) with details on that genetic variant is opened in a new window.

4.3.6 Organise Windows

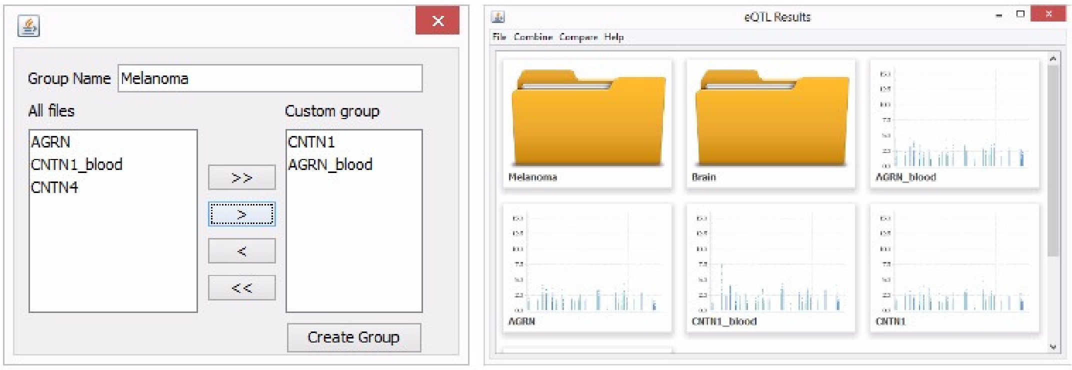
Users often wish to rearrange and resize tables and plots during analysis (e.g. to help identify patterns). In the software, individual windows are organised and moved on the display by clicking on the title bar and moving/dragging with the mouse.

4.3.7 Filter Data

With large data sets, filtering is essential to quickly find and work with a subset of data, for example, a user-defined significance threshold. Data sets are filtered in the plot or table windows by using a spinner to manipulate the significance threshold values (see Figure 4.2).

4.3.8 Combine Files

Gene expression files are typically explored independently. Therefore, patterns shared with other files are only identifiable after individually exploring each file. A clear improvement would be to combine gene expression files to examine patterns.



Group Name Melanoma

All files

AGRN
CNTN1_blood
CNTN4

Custom group

CNTN1
AGRN_blood

Figure 4.3: (left) Combination form. (right) Customised combination of files based on a disease, e.g. melanoma.

A combination of files is created by clicking on the Combine menu and choosing whether to customise a combination or to automate a combination of files based on a criterion, e.g. tissue. Choosing to automate a combination displays a folder in the grid view that is accessed as an independent file. Choosing to customise a combination prompts the user for a combination name and gene expression members (see Figure 4.3).

4.3.9 Match Significance across Files

The relationship between genetic variants can be derived by comparing a collection of these variants from one gene expression file with those from another file. Genetic variants collected from one gene expression file are compared against those from another file for significance across gene expressions. This is achieved by first saving a collection of genetic variants via a menu option. Then, another file is opened and the collection is compared against the file's data points via another menu option (see Figure 4.4).

4.4 Touch versus Mouse Input

Gene expression files are combined to detect patterns and compare significance in an eQTL study. To assert touch interaction as a baseline for comparison

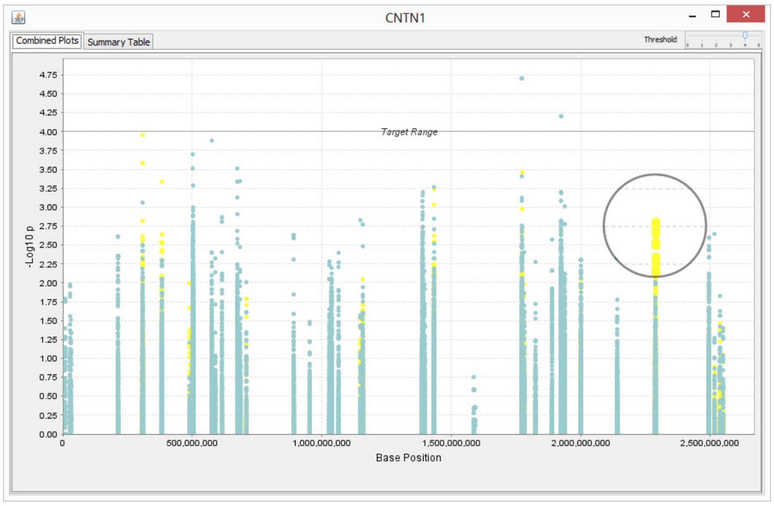


Figure 4.4: Genetic variants collected from one gene expression file are compared against the displayed file's data set. Matches are highlighted in yellow and magnified in the figure. These matches can potentially signify a complex association between the gene expression files compared.

against tangible interaction, this research compares touch and mouse input when combining gene expression files. The purpose of this study is to determine the strengths and weakness of touch interaction and the combination approach that needs to be taken into account when developing the baseline touch UI and the experimental task in Chapter 7. The GUI uses a selection and dragging approach for moving items into combinations (see Figure 4.3).

The results from previous research for combined selection and dragging actions are contradictory (see Section 2.3.3). Therefore, the main goal of the experiment described below is to determine the strengths and weakness of touch interaction for the combination approach utilised with the GUI. The experiment also compares the effect of two different font sizes, 12 pt and 20 pt, to examine the effect of size of contact on speed and accuracy. Three different list sizes are also considered to test the volume of items on display.

Based on the previous comparisons between touch and mouse inputs (see Table 2.6), we hypothesised that the touch input would outperform the mouse input in speed because touch interaction is both direct and natural (**H1**). The touch input would result in more errors than the mouse input because of possible occlusion and inaccurate contact (**H2**). We also hypothesised that the difference will be greater for a smaller font size (12 pt vs. 20 pt) (**H3**).

4.4.1 Fitts' Law

Fitts' law models human movement in HCI and ergonomics in order to predict the amount of time it takes to move to and select a target area [62]. While it was first used to predict tasks related to work efficiency (e.g. [258]), It has been used to model the act of pointing either by direct touch or a pointing device, e.g. mouse. The usual form of Fitts' law predicts that the time it takes to point to a target is is logarithmically related to the width of the target (target size) over the distance to the target:

$$MT = a + b \ \log_2 \left(\frac{2A}{W} \right) \tag{Fitts' law}$$

Where MT is the movement time. a and b are the intercept and slope of the linear regression model determined empirically. A and W are the distance to target and target width, respectively. The difficulty of the motor task is determined by the index of difficulty ($ID = \log_2 \left(\frac{2A}{W}\right)$), while the index of performance ($IP = \frac{1}{b}$) determines the human rate of information processing.

A large body of HCI research evaluates the performance of computer input devices for a multitude of tasks using completion time as a measure of performance (see Section 2.3.3), whereby Fitts' law and its variations (e.g. [151]) were occasionally used to compare different input devices. One of the earliest application of Fitts' law in HCI compared the performance of four devices in selection text on a CRT display, where one of the conditions was the newly unveiled mouse [31].

The fundamental rules of Fitts' law indicate that the larger the target and the closer it is the easier it is to point at via touch or pointing device. The law, nevertheless, has few limitations. The movement in Fitts' law describes a specific situation where the movement towards a target area is limited to one dimension, i.e. W refers to the width of the target and not its height or overall size. The process of pointing to a target is divided into an initial and final movement phases, where the initial movement encompasses the rapid acceleration towards a target and the final phase where the target is pointed at [78]. Whatever the size of the target area, the initial phase is likely to be similar, while the final phase was likely to be affected by virtual versus physical control.

Fitts' law also favours interaction that requires less tension (i.e. selection), despite the potential benefits of dragging. This behaviour better reflect everyday practices that involve moving an item from one point to another. Additionally, the randomisation of the order of items in the beginning of each sub task is not preserved. This is to overcome the learnt behaviour of over practiced controls, which is supported by Fitts' law.

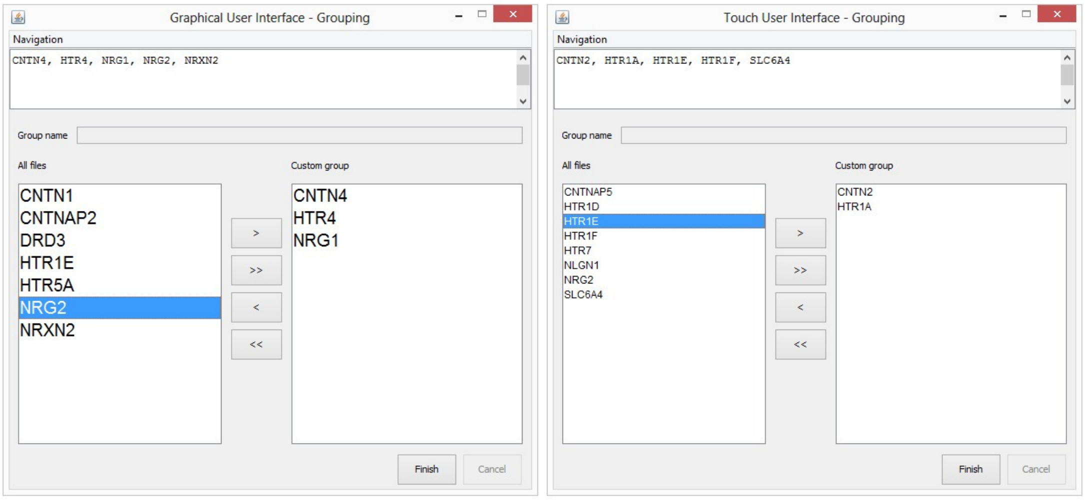
Within the context of this research, the experiment aims to determine the strength and weaknesses of touch interaction given a combination task (generalised from a specific eQTL task) for the purpose of developing a comparative baseline touch UI for eQTL visualisation. Selection and dragging options are presented to the user to determine preference and potential limitations of either approach. The experiment considers the size of the contact (determined by the height of the text) and number of items in a list. Although the use of a predictive model has many benefits, this experiment evaluated the performance of touch and mouse input using completion time to overcome some of the law's limitations.

4.4.2 Method

A within-participants design was used, where each participant took part in all conditions. The experiment had three factors: input (mouse and touchscreen), font size (12 pt and 20 pt) and list size (10, 15 and 20 items).

4.4.2.1 Participants

Twelve participants (10 females and two males) with a mean age of 24.17 years ($SD = 5.83$) took part. Each participant was randomly allocated to one of the four input × font groups, which counterbalanced the order of participants used in the interface and the order in which the font sizes were presented while the list size was randomly determined. All participants gave informed consent and were paid for their participation.



CNTN4, HTR4, NRG1, NRG2, NRXN2

CNTN1
CNTNAP2
DRD3
HTR1E
HTR5A
NRG2
NRXN2

CNTN4
HTR4
NRG1

CNTN2, HTR1A, HTR1E, HTR1F, SLC6A4

CNTNAP5
HTR1D
HTR1E
HTR1F
HTR7
NLGN1
NRG2
SLC6A4

CNTN2
HTR1A

Figure 4.5: (left) The combination task with 20 pt font. The combination items are enumerated at the top. The all-files list displays a list of items from which the participant selects the five target items, one at a time to move them to the custom-combination list. (right) The combination task with 12 pt font.

4.4.2.2 Materials

The combination task was isolated from the GUI and ran on an HP Pavilion Sleekbook 15-b160ea 15.6-inch touchscreen laptop. The resolution of the display was 1366 × 769 pixels with a seven ms touch response time. For the mouse input, the participants interacted with the vertically positioned screen while seated. For the touch input, the laptop touchscreen was placed horizontally in front of the participants to counter fatigue caused by vertically interacting with a touchscreen (the gorilla-arm effect).

The size of investigated gene expressions in research is varied with values ranging from very little (e.g. Welsh et al. [260] analysed two types of tissues) to much more (e.g. Chen et al. [35] examined 46 genes). Five item combinations were deemed suitable for this experiment as its a decent sized combination, but still allows for future scalability. In the task, the participants had to select five items from one list and move them to another list to create a combination.

At the beginning of each trial, a participant was presented with two lists: one empty list and another list that displayed 10 to 20 gene expression files. The number of gene expression files presented were chosen to reflect the potential need for scrolling. The sizing of the font ranged from 12pt (16px) and 20pt (26.67px). Current guidelines for touch devices recommend minimum target sizes ranging from 26 pixels to 44 pixels [104, 176, 263], approximately 20pt to 33pt for textual targets. Therefore, this range reflected the average sizes found in GUI and the minimum recommended size for touch interaction. The five target items were also presented to the user at the top of the window (see Figure 4.5).

The participant's task was to first find the items in one list and move them to the combination list. Items were moved from one list to another by either clicking on the appropriate button or by dragging the item to the combination list. The only feedback provided by the software was when the participant attempted to move on to the next trial prior to correctly forming the combination. When a mistake was made, the participants were informed and prompted to rectify it.

4.4.2.3 Procedure

The participants individually performed the tasks. The experimenter first demonstrated how to use the software for one condition (e.g. mouse input), and the participant performed six practice trials, one for each combination of font and list size. This allowed familiarisation with the interface and task. Next, the participant performed nine test trials for one font size and three trials for each list size presented in a random order (e.g. 12 pt as shown in Figure 4.5). This was then followed by nine more trials for the other font size (e.g. 20 pt) with three randomly ordered trials for each list size. The entire process was then repeated for the other input device (e.g. touch input). All interactions were recorded in a log file.

4.4.3 Results

The results were analysed using an analysis of variance (ANOVA) with the input type (mouse versus touch), font size (12 pt versus 20 pt) and list size (10, 15, 20 items) as repeated measures. Only significant interactions are reported.

The trial duration was measured as the time between the start of a trial and the successful formation of a combination. The following analyses used a

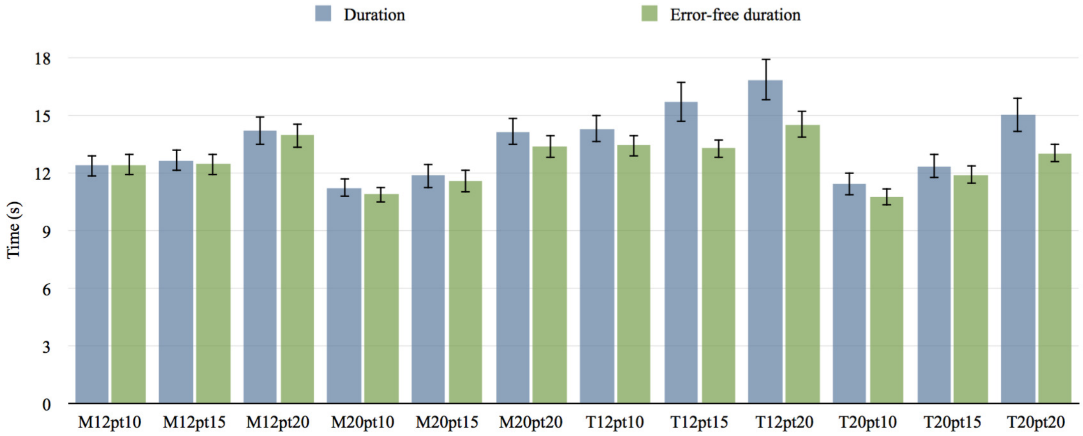


Figure 4.6: Mean duration for all trials and error-free trials. Error bars show the standard error of the mean (SE).

participant's average duration for each combination of input type, font and list size. An ANOVA showed that the participants were significantly faster with the mouse input than with the touch input ($F_{1,35} = 9.86, p < .01$), significantly faster with the 20 pt font than with the 12 pt font ($F_{1,35} = 21.73, p < .01$) and significantly faster with lists of 10 items ($F_{1,35} = 48.30, p < .01$). There was also a significant input × font interaction ($F_{1,35} = 5.52, p = .03$). The mouse input at 20 pt took the least time to complete the trial followed by the touch input at 20 pt, while the touch input at 12 pt took the longest time to complete.

A repeated ANOVA that only included the error-free trials showed that input types did not have an effect on task duration, while font and list sizes had a significant effect ($F_{1,35} = 20.09, p < .01$ for font size and $F_{1,35} = 49.39, p < .01$ for list size) with the task taking longer to complete with the 12 pt font and the 20 item list (see Figure 4.6). This indicates that the main effect of input types was owing to the participants making errors.

Five sources of errors were logged for analysis:

- Selection: selecting an item that is not a member of the combination.
- Drop: an item is dropped back to the item list.
- Return-one: clicking on < button.
- Return-all: clicking on << button.
- Incomplete: attempting to finalise a combination when incomplete.

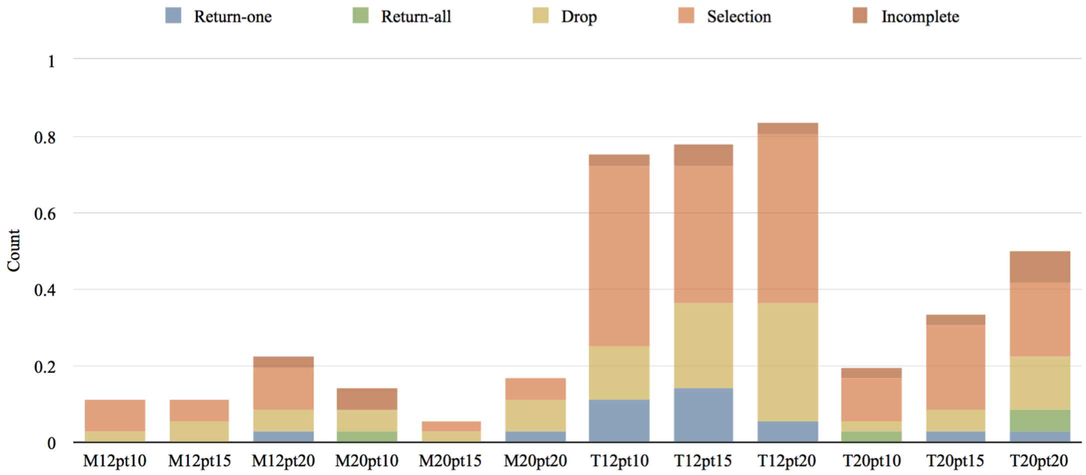


Figure 4.7: Mean return-one, return-all, drop, selection and incomplete errors for all trials.

An ANOVA of the five sources of errors showed that the participants made significantly more mistakes with the touch input than with the mouse input ($F_{1,35} = 21.74, p < .01$) and significantly more mistakes with the 12 pt font than with the 20 pt font ($F_{1,35} = 13.84, p < .01$). There was also a significant input × font interaction ($F_{1,35} = 13.84, p < .01$) with the greatest number of mistakes recorded for the touch input at 12 pt font. Figure 4.7 illustrates the mean of the five error types for each input type × input times × list size.

4.4.4 Discussion

The main goal of the experiment was to compare the performance of two input types (mouse and touch), two font sizes (12 pt and 20 pt) and three list sizes (10, 15, and 20 items) for the task of forming a combination. In terms of both speed and accuracy, the mouse input outperformed the touch input; it took participants longer to form combinations while also incurring more errors than with the touch input. This section discusses these findings in the context of previous work (see Table 2.6) and our hypotheses regarding speed (**H1**) and accuracy (**H2** and **H3**).

The participants took longer to perform combinations with the touchscreen compared with the mouse. The speed results mirrored those of [165], where touch performed significantly worse than other direct and indirect devices in a drawing task which used selection and dragging actions. Therefore, there was no evidence to support hypothesis **H1**, which postulated that the touch input would outperform the mouse input in speed. However, further analysis of error-free trials found no significant effect for input type, indicating that the underlying cause of the speed difference was the errors that the participants made using touch input.

The inaccuracy of the touch input was consistent with previous research [41, 67, 99, 202]. This supports **H2**, where inaccuracies were expected with the touch input more often than with the mouse input. This is likely to be attributed to the occlusion of the hands when interacting with the touchscreen and inaccurate contact with target items owing to target size or placement. The effect of target size on duration and accuracy has been previously explored [41, 60, 67, 202, 208], where larger target sizes reduced inaccuracies and duration for the touch input. Font size, i.e. target sizes, showed a significant effect on the duration of the combination task, where selection and drop errors were reduced with larger target sizes (see Figure 4.7). This provides evidence for **H3**.

The participants were given two techniques to move an item from one list to another after selection: clicking a button to move an item back and forth or dragging an item from one list and dropping it onto another. Interaction technique preference was extracted from the log files, which showed a preference for drop and drag at a little over 80% for the mouse input. The participants utilised the button technique for more than 60% of their interactions with the touch input. This shows that despite the reduced utilisation of the drag and drop technique for the touch input, the drop errors were still significantly higher. This difference is also represented in the variations between error-prone and error-free computations, particularly for the 12pt font trials. .

In all trials, the participants were asked to form five item combinations from various list sizes. The list size (10, 15 and 20 items) had a significant effect on duration. In a list of unfamiliar names, as is the case with gene names, the participants scanned the shorter lists faster than the longer lists. In longer lists, the participants made significantly more mistakes when dragging and dropping items between lists.

The results of these experiments considerably influenced how this research approached the design of the combination task interaction for the touch UI (see Chapter 5) and the experimental task comparing the touch UI and TUI (see Chapter 7). An ideal combination task for the purpose of using the touch UI as a baseline for comparing touch and tangible interaction would reduce inaccuracies and in turn duration by limiting drag and drop actions and reducing the number of files being explored for combination patterns.

4.5 Summary

In this chapter, the methods undertaken to elicit eQTL analysis requirements are described and their application in a GUI is explored. Semi-structured interviews were conducted to explore the domain of quantitative genetics and to elicit interface requirement for an eQTL visualisation application (see Section 4.2). From these interviews, design implications were extracted and along with the analysis scenarios were formulated into nine functional requirements (see Table 4.2). The requirements were implemented in a GUI to explore the eQTL tasks and their potential interactions. To assert touch interaction as a baseline for comparison against the TUI, an experiment compared touch and mouse input for the eQTL task of combining files (see Section 4.4). The results showed that touch accuracy is affected by the size of the contact point, list size and drag and drop interaction. These factors are taken into account when designing the touch UI (see Section 5.2) and the experiment's task used to compare touch UI and TUI (see Chapter 7).

Chapter 5

Infovis Touch UI and TUI Design

5.1 Introduction

As it is outlined from the interviews and analysis scenario (see Section 4.2), three levels of data abstraction are manipulated to identify areas and genetic variants of interest: independent gene expression files, combined gene expression files, and collections of genetic variants. Table 5.1 lists the eQTL tasks and their respective subtasks. Figure 5.1 illustrates the basic design environment with a planar display that may be used with the touch UI and TUI systems. The display is divided into a viewing area, where a user can open and interact with gene expression windows, and a file explorer, where a series of uploaded gene expression files are displayed as thumbnails. A user would be able to scroll through the file explorer to scan and expose additional files. The gene expression files' thumbnails should be large enough for a user to be able to discern patterns and determine potential relationships.

The remainder of this chapter explores the design options for developing a touch UI and a TUI for interactive infovis. First, the design options for a touch UI are explored with gesture categories and handedness. The design options for an exclusively tangible TUI with an interactive surface are also examined. Objects utilised with the TUI are probed along a continuum of objects' meanings. Design options that are considered for the touch UI and TUI systems are then combined to form a hybrid interactive surface TUI.

eQTL task	Subtask
Open/close files	Open files Hide files Close files
Switch windows	
Scroll, pan and zoom	Scroll Pan Zoom
Select genetic variants	Select genetic variants Expose information Deselect genetic variants
Access external sources	Open external sources Close external sources
Organise windows	Translate windows Rotate windows
Filter data	
Combine files	Add to combination Remove from combination
Match significance across files	

Table 5.1: eQTL tasks and subtasks.

5.2 Touch UI Design Options

Touch UIs have gained ground due to the widespread use of multi-touch smartphones. Using various touch pointing technologies [206], touch UIs are able to simultaneously detect direct touch input from multiple fingers, accommodating usage with one or two hands. This form of interaction is intuitive and natural. For instance, a user may display and interact with a file on the touch surface with one hand, while using the other hand to clear other displayed files. Such behaviour is a digital facsimile of real-world interactions.

Surface gestures based on user behaviour are classified along four dimensions: form, nature, binding and flow [264]. This classification describes the gesture design space and the findings indicate that users generally prefer one-handed

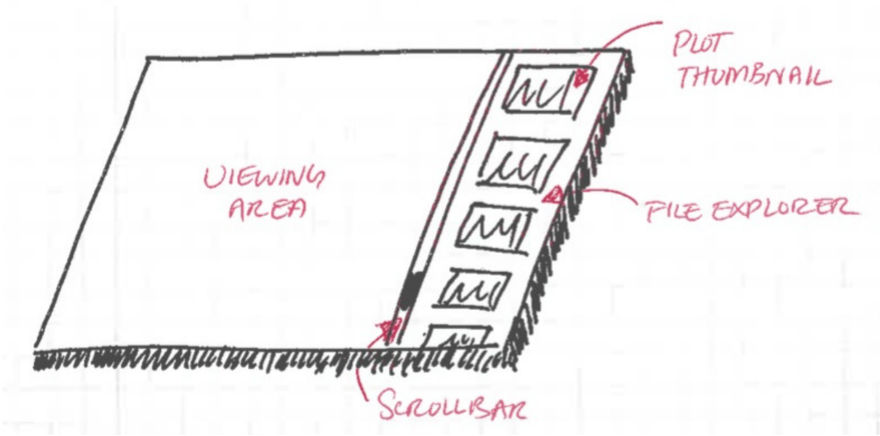


Figure 5.1: The interactive surface with a viewing area and a file explorer. The file explorer displays a list of plot thumbnails representing gene expression files and the viewing area is used to view and interact with the files.

gestures using one or more fingers and that desktop idioms influence users' gesture preference. Gestures may also be categorised based on degrees of freedom, spatial occurrence, semantics, trajectory complexity, number of users, number of fingers and timing [39]. The gesture classes may also be expanded to include number of hands properties. For the purpose of this thesis, we extract classification and properties from the two frameworks [39, 264] to describe the touch UI design options.

The design options for the touch UI are first divided into two larger sets: desktop idioms and gestures. Desktop idioms utilise the menu and form-based techniques that are popularly used in desktop setups and GUI. However, the controls are adapted for the purpose of touch sensing (e.g. button size restrictions). Gestures are one-handed or two-handed interactions that differ in nature (metaphorical or abstract) and flow (discrete or continuous). One-handed gestures may be completed using a single finger or multi-fingers where two or more fingers are utilised to interact with the surface. Two-handed interaction techniques, i.e. bimanual, employ both hands to provide intention for manipulation. The gestures considered are either abstract or metaphorical with continuous and discrete flows. While there are other dimensions (form and binding) and categories that can be considered, the options are limited to simplify description.

Gesture	Name	Description
	Tap	Briefly touch surface with fingertip.
	Double tap	Rapidly touch surface twice with fingertip.
	Drag	Move fingertip over surface without losing contact.
	Five finger pinch	Touch surface with five fingers and move them closer together.
	Press	Touch surface for extended period of time.
	Rapid swipe	Quickly brush surface with fingertips. The swipe distance determined by number of fingers used.
	Swipe (left and right)	Quickly brush surface with fingertips.
	Swipe (up and down)	
	Pinch and spread	Touch surface with two fingers and bring them closer together (pinch) or further apart (spread).
	Asynchronous bimannual pinch and spread	
	Synchronous bimanual pinch or spread	
	Rotate	Touch surface with two fingers and move them in a clockwise or counterclockwise direction.
	Asynchronous bimanual rotate	
	Synchronous bimanual rotate	
	Asynchronous press and drag	Press with one finger and move other finger over surface without losing contact.

Table 5.2: Gestures utilised in the touch UI design options are described here for reference.

Table 5.2 describes the gestures utilised in the design options described for the touch UI. Table 5.3 summarises the eQTL task requirements and the design options for touch UIs given the two main categories of desktop idiom and gesture (abstract and metaphorical). The following sections describe and illustrate these

eQTL subtask	Desktop idiom	Gesture	
		Abstract	Metaphorical
Open files	☰	👆 | 👆	👆
Hide files	☰		👆
Close files	☰		👆
Switch windows	☰	👆	✋
Scroll	▣		👆 👆
Pan	☰		👆 👆
Zoom	☰	👆	👆 | 👆 👆 | 👆 👆
Select genetic variants		👆	
Expose information		👆	
Deselect genetic variants		👆	
Open external sources	☰		👆
Close external sources		👆	👆
Translate windows			👆
Rotate windows			👆 | 👆 | 👆
Filter data	⫶ | ▭ | ▭		👆
Add to combination	☰		👆 👆
Remove from combination	☰		👆 👆
Match significance across files	☰		👆

Table 5.3: eQTL tasks and touch UI design options using desktop idioms, abstract and metaphorical gestures.

design options for each eQTL task, where nature, flow and handedness of the interactions are also identified.

5.2.1 Open/Close Files

A set of statistically analysed gene expression files could be loaded and displayed as thumbnails in the file explorer. For the purpose of investigating a single gene expression file, the interface needed to support three subtasks:

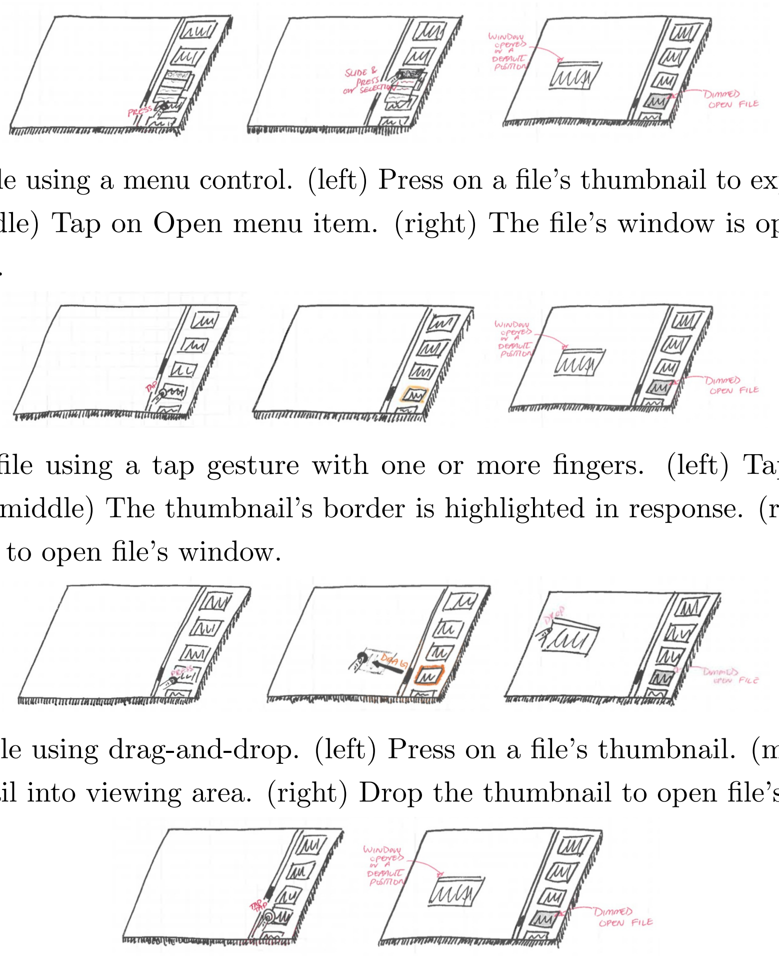
- Open a file to view in the viewing area.
- Close a file.
- Hide a file from view.

A gene expression file could be opened using the high-level categorisations of desktop idiom and one-handed gestures. A user would open a gene expression file by opening a context menu in the file explorer. To expose the context menu, the user would press on a gene expression file's thumbnail using one or more fingers for a certain period of time. A series of options would be presented to the user, including the option to open a file. The user would use an abstract tap gesture on the Open menu option to open the file. This would open the file's windows in a predetermined position in the viewing area, e.g. the centre of the surface (see Figure 5.2a).

A user could also open a file using one-hand with single or multi-finger gestures (an abstract and discrete gesture). First, the user would tap on a gene expression file's thumbnail in the file explorer. The thumbnail would become highlighted in response to the interaction. Within a certain timeframe and if the user taps with one or more fingers anywhere on the viewing area, the gene expression file would open in a window (see Figure 5.2b).

An alternative approach using single or multiple fingers could use a continuous metaphorical drag-and-drop. A user would first press on a file's thumbnail in the file explorer and the thumbnail would respond with a highlighted border. The user would then drag the thumbnail from its initial position and drop it in the viewing area as is shown in Figure 5.2c.

It could also be possible to open a gene expression file with another abstract and discrete gesture, double tap. A user would double tap on a file's thumbnail using one or more fingers. This would open that file's window in a predetermined position in the viewing area (see Figure 5.2d). Once a gene expression file was opened, its thumbnail would be dimmed to disable any further interactions.


PRESS
SLIDE & PRESS ON SELECTION
WINDOW OPENED IN A DEFAULT POSITION
DIMMED OPEN FILE
PRESS
DRAG
DROP
TAP TAP

(a) Open a file using a menu control. (left) Press on a file's thumbnail to expose context menu. (middle) Tap on Open menu item. (right) The file's window is opened in the viewing area.

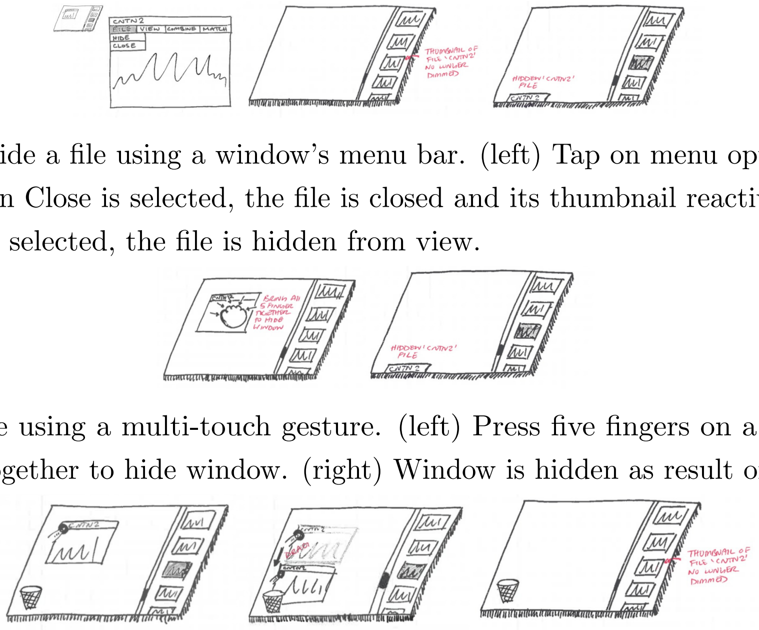
(b) Open a file using a tap gesture with one or more fingers. (left) Tap on a file's thumbnail. (middle) The thumbnail's border is highlighted in response. (right) Tap in viewing area to open file's window.

(c) Open a file using drag-and-drop. (left) Press on a file's thumbnail. (middle) Drag the thumbnail into viewing area. (right) Drop the thumbnail to open file's window.

(d) Open a file using a double tap gesture. (left) Double tap on a file's thumbnail. (right) The file's window is opened in the viewing area.

Figure 5.2: Design options for opening a gene expression file in a touch UI.

Each of the four options considered for opening a gene expression file via touch present advantages and disadvantages to analysts that are typically conducting work on a conventional workstation using a mouse and keyboard. The use of a context menu via touch requires the user to perform multiple presses and taps to open a single file, which could prove repetitive when opening consecutive files. The options to tap twice, once on the file explorer and then the viewing area, could be interrupted and cause a user to lose their line of thought. The drag


CNTN2
FILE
VIEW
COMBINE
MATCH
HIDE
CLOSE
THUMBNAIL OF FILE 'CNTN2' NO LONGER DIMMED
HIDDEN 'CNTN2' FILE
CNTN2
BRING All 5 FINGER TOGETHER TO HIDE WINDOW
CNTN2
HIDDEN 'CNTN2' FILE
CNTN2
CNTN2
DRAG
CNTN2
THUMBNAIL OF FILE 'CNTN2' NO LONGER DIMMED

(a) Close or hide a file using a window's menu bar. (left) Tap on menu option to select. (middle) When Close is selected, the file is closed and its thumbnail reactivated. (right) When Hide is selected, the file is hidden from view.

(b) Hide a file using a multi-touch gesture. (left) Press five fingers on a window and bring them together to hide window. (right) Window is hidden as result of the gesture.

(c) Close files using a dedicated area. (left) Press on a file's window menu bar. (middle) Drag the window to recycle bin. (right) The window is closed and file's thumbnail reactivated.

Figure 5.3: Design options for hiding or closing a gene expression file in a touch UI.

and double tap options present the least effort and could reduce interruptions. Nevertheless, drag and drop allows the user more control over the placement of the file and could potentially support visual thinking.

A gene expression window could be closed or hidden using a desktop idiom or multi-touch gestures. To close or hide a gene expression file's window, a menu bar could be placed at the top left corner of the file's window with Close and Hide menu options. A user would be able to hide a window by tapping on Hide using one or more fingers. This would minimise the window from view with the possibility of later retrieval. Similarly, a user could close a window by tapping on Close. This would close the file's window and its thumbnail would reset its dimness to reactivate interaction (see Figure 5.3a). Alternatively, the context menu described when opening a window (see Figure 5.2a) could also be used to close or hide a file's window from view.

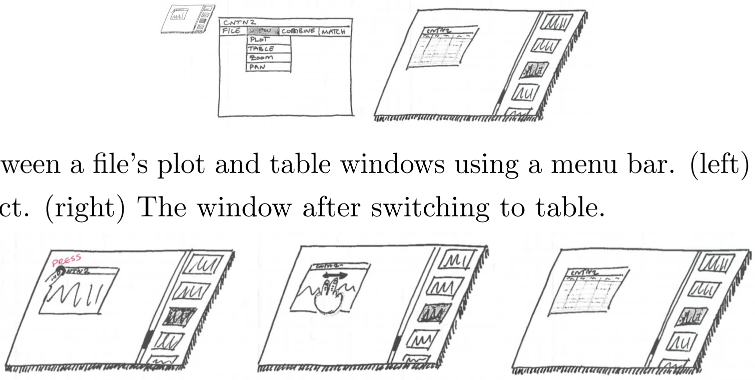
A gene expression window could also be hidden using continuous and metaphorical multi-touch gestures. For example, a user would press with all five fingers on the file's window and then bring them close together to hide the window (see Figure 5.3b). Alternatively, for closing files, an area could be dedicated for that purpose. For example, a graphical recycle bin icon would be placed on the bottom left corner of the viewing area. A user would press on a window's menu bar using one or more fingers, drag the window and then drop it in the recycle bin (see Figure 5.3c).

Of the two interactions considered for closing a gene expression file, a Close option from a menu bar or a drag and drop to a graphical icon, the drag-and-drop option would allow for continuous uninterrupted interactions compared to tapping on a Close option. While the same rationale could be applied for the two interactions considered for hiding a gene expression file, a Hide option from a menu bar or a metaphorical gesture, the use of an unfamiliar metaphorical gestures could potentially cause confusion. This is primarily due to analysts' familiarity of conventional setups and interactions.

5.2.2 Switch Windows

eQTL analysis results could be viewed in tables or visualised in Manhattan plots, with users switching between views during the interpretation process. A desktop idiom or an abstract or metaphorical gesture could be used to switch between view windows. To switch between table and plot windows, a menu bar could be anchored to the top left corner of the file's window with Plot and Table menu options. A user would be able to select a window to view by tapping on Plot or Table using one or more fingers. This would adjust the window's view (see Figure 5.4a).

Alternatively, a user could switch between windows using abstract or metaphorical one-handed gestures. For example, a user could press on a window's title bar for a certain period of time to flip between windows. Similarly, multi-finger techniques could also be used to switch between windows. A user would use multiple fingers to press on the window and then swipe to switch views (see Figure 5.4b).



(a) Switch between a file's plot and table windows using a menu bar. (left) Tap on menu option to select. (right) The window after switching to table.

(b) Switch between windows using one or more fingers. (left) Single finger press and (middle) multi finger swipe. (right) The window after switching to table.

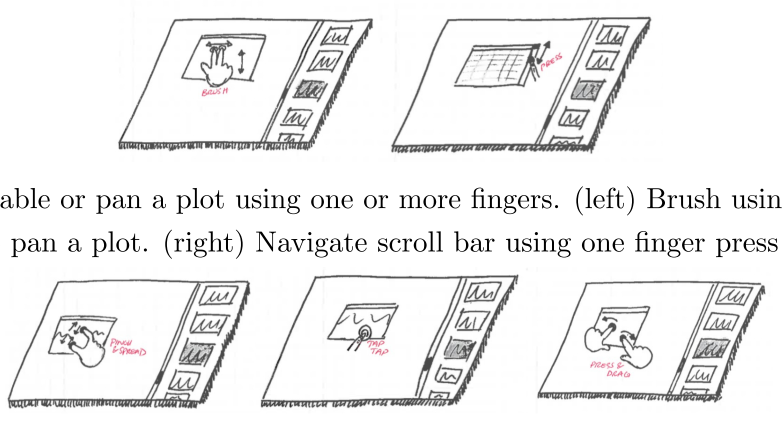
Figure 5.4: Design options for switching between a gene expression file's windows in a touch UI.

The option to view table and plot views simultaneously could also be presented as a viewing option. Similar to the interaction options described for viewing a table or plot window, a user could use a menu bar to choose a Plot/Table menu option. Alternatively, the switch to a simultaneous plot and table view could be achieved by either pressing on a window's title bar where the views would follow a certain order. For example, a plot would be used as the default view where a first press on a window's title bar would change the view to a table and another press would change the view to a simultaneous plot and table view. This would similarly be the case for swiping with a gesture to switch between view.

For switching views in gene expression files, desktop idioms and gestures are considered. The reliance of users' on desktop idioms strengthens the option of using a menu bar with a Table, Plot and simultaneous view options to choose from. Another strength of using a menu bar could be the ability to combine a number of options onto one control, which would be beneficial when the number of views increases. Despite the abstract nature of the press gesture considered, it would be easy to perform and potentially recall. The naturalness of the swipe gesture parallels that of flipping a page, an everyday interaction both familiar and intuitive. Nevertheless, the complexity of both gestures would increase as the number of view options expand.

5.2.3 Scroll, Pan and Zoom

Interactively manipulating plots and tables could allow users to interact with more information that is not conveniently displayed at one time. Plots and tables could be navigated by panning and zooming to alter a user's viewpoint and increase their understanding of the results. In a table window, hundreds of thousands of rows would represent the genetic variants contained within a data set. To ease scrolling, vertical and horizontal scrollbars could be displayed for a user to manipulate. To scroll vertically or horizontally, a user would press the scrollbar thumb using one or more fingers and drag to navigate. To allow for faster scrolling a user could adopt a continuous metaphorical gesture; the user would brush using one or more fingers on the surface of the window in the opposite direction of the scrolling (see Figure 5.5a).



(a) Scroll a table or pan a plot using one or more fingers. (left) Brush using more than one finger to pan a plot. (right) Navigate scroll bar using one finger press and drag.

(b) Zoom a plot using one or two hands. (left) Pinch and spread on a plot window to zoom out and in. (middle) Double tap on a plot to zoom. (right) Pinch and spread on a window using two hands symmetrically.

Figure 5.5: Design options for scrolling, panning and zooming a gene expression file's table or plot window in a touch UI.

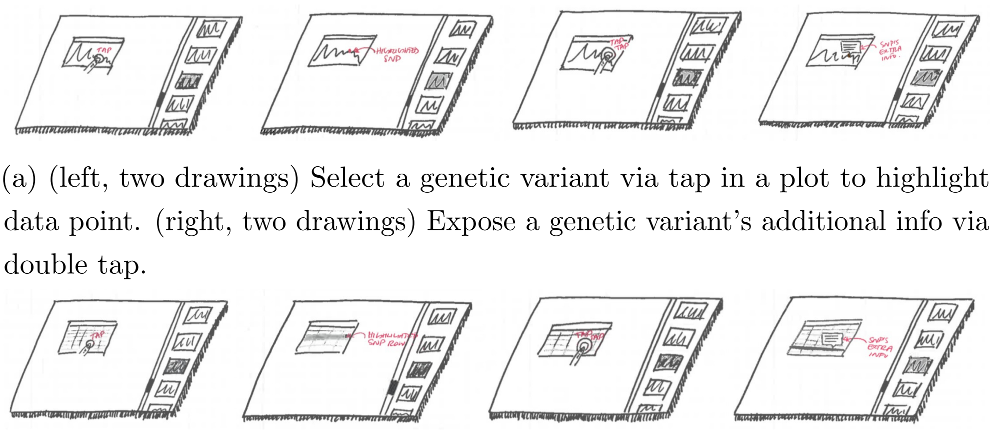
A user's viewpoint could be shifted in a plot window horizontally or vertically via a metaphorical panning gesture. Similar to scrolling, a user would brush using one or more fingers on the surface of the window in the opposite direction of panning. With scrolling or panning, the number of fingers could be used to determine the distance of the pan or scroll. The windows would pan or scroll faster when more fingers are used (see Figure 5.5a).

Plot windows could further be navigated using various one-handed or two-handed metaphorical and abstract gestures to zoom in and out. A user would adopt a pinch gesture by touching the plot's window with two fingers and bringing them close together to zoom out. Inversely, a user would move the fingers apart to zoom into view (continuous and metaphorical). Alternatively, a user could zoom in and out in preset increments by tapping twice on the surface of the window (discrete and abstract). Using two hands asymmetrically, a user could zoom by pressing at a certain point on the plot with one hand then using the other hand to drag inwards or outwards. It could also be possible to press and drag with both hands symmetrically to zoom (see Figure 5.5b).

A gene expression file's table view could be manipulated using scrollbars or a metaphorical gestures to expose additional rows. Both techniques are usually adopted in desktop applications and touch UIs. In the case of touch UIs, scrollbars and brushing are commonly supported together. Visualising gene expression data sets allows the user more navigational options, e.g. panning and zooming. Despite the uniform purpose of panning a plot and scrolling a table, the unrestrained nature of panning (scrolling would typically only require vertical and horizontal sweeps) would be best supported with a brushing gesture. The use of pinch and spread gestures to zoom in and out of a plot are well-established in touch UIs and present a more attractive option compared to using an abstract gesture.

5.2.4 Select Genetic Variants

Genetic variants (SNPs) could be selected in plots or tables to highlight their importance or to display additional information relating to that genetic variant. Each genetic variant should be plotted against its significance in a Manhattan plot. Single-finger abstract gestures could be used to select genetic variants. A



(a) (left, two drawings) Select a genetic variant via tap in a plot to highlight data point. (right, two drawings) Expose a genetic variant's additional info via double tap.

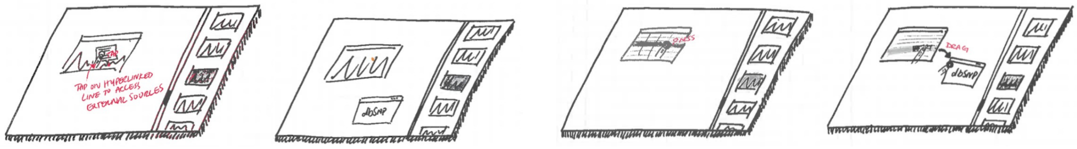
(b) (left, two drawings) Select a genetic variant via tap in a table to highlight row. (right, two drawings) Expose a genetic variant's additional info via double tap.

Figure 5.6: Design options for selecting genetic variants in a touch UI.

user would tap the surface of a genetic variant using a single finger to highlight it for later collection or exploration. To view additional information, a user could tap the genetic variant twice with a single fingertip (see Figure 5.6a). Similarly, in a table window, a user could tap on a row with a single finger to highlight it. A user would display additional information about a genetic variant by tapping twice on a row (see Figure 5.6b). In either plot or table windows, the user would be able to deselect a genetic variant by tapping on its surface again. The naturalness and ease of tapping to select a row or data point in a touch UI would exemplify the benefits of touch over conventional set ups.

5.2.5 Access External Sources

External data sources could be accessed when exploring eQTL to gain prior knowledge, endorse findings or provide explanation for the significance of a gene or a genetic variant. Access to various sources while exploring the analysis results should facilitate interpretation and understanding. External data sources could be accessed via links included with the infotip, which are exposed when a user selects a genetic variant with a discrete abstract gesture (see Section 5.2.4). The user would tap on a hyperlink to open a window displaying information about



(a) Open external sources using a tap gesture. (left) Expose a genetic variant's information via tap on a linked source. (right) A window with the data source is opened in the viewing area.

(b) Access external data sources using drag-and-drop. (left) Press on a genetic variant's row. (right) Drag a genetic variant's row to the viewing area to open a data source's window.

Figure 5.7: Design options for accessing external data sources in a touch UI.

the selected genetic variant from the linked data source (see Figure 5.7a). Access could also be possible using a metaphorical drag-and-drop gesture. A user would drag a data point from a plot window (or a row from a table window) and drop it onto the viewing area to open and display an external database window (see Figure 5.7b). An external source window would be closed via drag-and-drop to the recycle bin or by tapping on close in the window.

The options considered for accessing external data sources present challenges and opportunities to analysts that typically conduct their work on conventional setups. The use of hyperlinks would be expandable (e.g. include access to multiple databases using multiple hyperlinks) and its interaction (tap on hyperlink to open an external window) is customary. The drag and drop alternative considered is an attractive options as it would give users more control over the placement of the external data source's window and could potentially support spatial thinking. Nevertheless, this approach could increase the error rate for touch (e.g. [41, 67]).

5.2.6 Organise Windows

Users would often wish to view multiple files at the same time, which could be rearranged during analysis interpretation (e.g. to help identify patterns shared among opened files). To move a window, a user could use a continuous metaphorical gesture using one hand. The user would press on the file window's title bar and then move it to translate its position in the viewing area. A user could also rotate a window using a one-handed multi-finger continuous gesture. The user

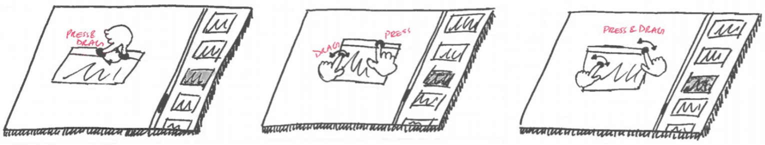


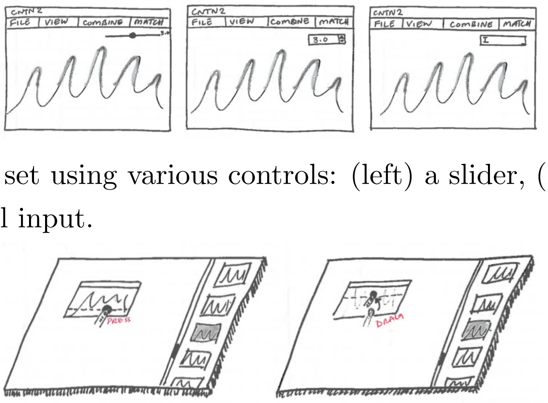
Figure 5.8: Rotate a window using one or two hands. (left) Press with two fingers from one hand and move both fingers to rotate. (middle) Press with two hands and drag one hand to rotate window asymmetrically. (right) Press with two hands and drag both hands to rotate window symmetrically.

would press with two fingers on the window and move them in a clockwise or counter-clockwise direction. Using two hands, this technique could also be achieved symmetrically using a finger from each hand. Likewise, with two hands, a user could asymmetrically rotate the window. The user would press with one hand and rotate with the other to change the window's orientation (see Figure 5.8). The options considered for translating and rotating a gene expression window are well-established and familiar in touch UI as they imitate physical real-world interactions. Support for one- and two-handed rotations would be provided simultaneously as they serve variable needs.

5.2.7 Filter Data

Users, when working with large data sets, would often find filtering essential to quickly find and work with a subset of the data. For example, a user could decide to work with data that meet a user-defined significance threshold. Various desktop controls could be attached to a gene expression window (as a table or plot) to filter the file's data set (see Figure 5.9a). A user could adjust a slider by navigating its handle metaphorically via touch to adjust the significance threshold. Alternatively, a user could input a threshold value using a touch-sensitive number keyboard. In this case, input should first be verified. A user could also use a drop-down list to select a significance threshold via an abstract tap gesture (each gene expression file should have its own significance range given its maximum and minimum significance).

Using an interactive threshold line, data could also be filtered in a plot window. A user would press on the threshold line and drag it vertically to filter the data



(a) Filter a file's data set using various controls: (left) a slider, (middle) a drop-down list and (right) textual input.

(b) Adjust threshold line in a plot window. (left) Press on the threshold line. (right) Drag the threshold line to adjust.

Figure 5.9: Design options for filtering a gene expression's data set in a touch UI.

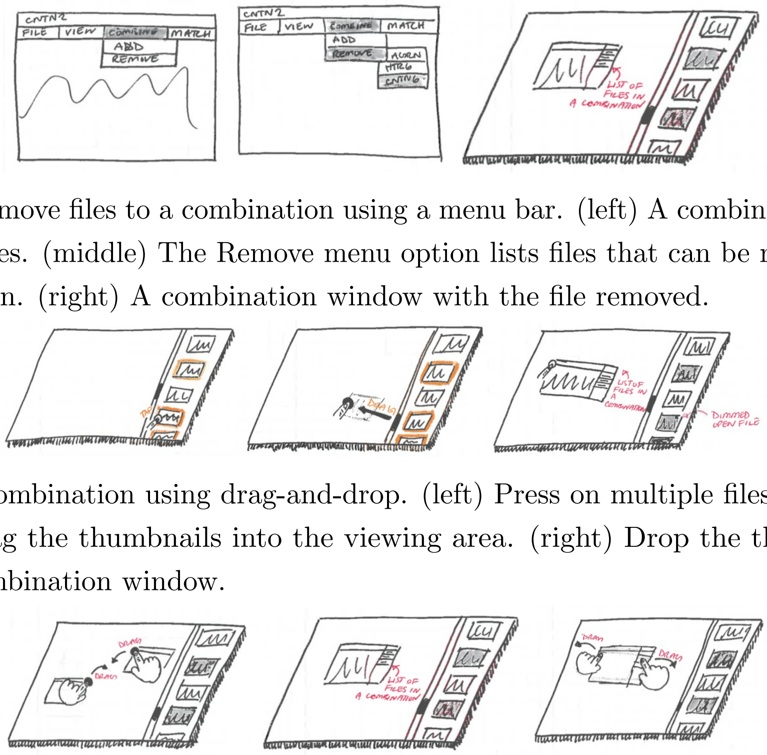
(see Figure 5.9b). In a plot window, filtering would dim non-significant data points. Table rows that do not meet the threshold requirement would be collapsed from view in a table window. An alternative metaphorical two-handed touch gesture could also be used to filter a data set in a plot window. A user would drag a semi-permeable filter [199] across a data set by using two hands on either sides of the filter barrier. Points that match the filtering criteria would be pulled along, whereby those that do not meet the criteria are not moved.

Various desktop controls and gestures are considered for filtering data sets in a table or plot. Desktop controls are common tools used for eQTL visualisation and analysis tools (e.g. [14]), and due to their familiarity could be easily adopted. Gestures offer rich alternatives that would mimic direct interactions. Dragging a threshold line using one-hand or a semi-permeable filter using two hands would simulate filtering behaviour that are aided with visual response. Additionally the behaviour of a semi-permeable filter could be an indicator of how much data has been filtered. Since a single variant is considered in the current touch prototype, the use of gesture could potentially be easier to understand and require less energy. Given the typical size of a genomic data set, manipulating a threshold line would potentially be simpler than using a semi-permeable filter.

5.2.8 Combine Files

Typically, gene expression files would be explored independently and patterns shared with other files are only apparent when exploring each file separately. Combining gene expression files could allow for the identification and examination of patterns from multiple files at the same time.

Desktop idioms or unique gesture could be used to form a combination of gene expression files. Menus could be used to form combinations, by including an option to add and remove gene expression files to/from a combination. A user would first tap on the Combine menu option via tap to display the options to



(a) Add or remove files to a combination using a menu bar. (left) A combination window with three files. (middle) The Remove menu option lists files that can be removed from a combination. (right) A combination window with the file removed.

(b) Form a combination using drag-and-drop. (left) Press on multiple files' thumbnails. (middle) Drag the thumbnails into the viewing area. (right) Drop the thumbnails to open the combination window.

(c) Combine two files' windows using two hands. (left) Press and drag the windows towards each other. (middle) The windows merge to form a combination window. (right) Press the title bar with one hand and press on a file from the combination window's side bar with the other hand, then drag apart to remove the file from the combination.

Figure 5.10: Design options for combining gene expression files in a touch UI.

either add or remove a file from a combination. The Add option would include all the files maintained in the session and the Remove option would include the files in the combination. To add to a combination, the user would select the file to be added via a discrete abstract tap gesture. The file's window would respond to this interaction by superimposing plots and merging tables. The window would also include a side bar with the names of the files forming the combination (see Figure 5.10a). Similarly, to remove a file, a user would select the file to be removed. This would remove that file's data set from the plot and table windows.

Combinations could be formed using multiple discrete abstract tap gestures. A user would first tap on thumbnails in the file viewer. The thumbnails would become highlighted in response. Within a certain timeframe, the user would tap anywhere on the viewing area to open a gene expression window that combines the selected files' data sets. Alternatively, single or multiple fingers could be used in a discrete combined abstract and metaphorical gesture. A user would select files to be combined from the file explorer via abstract tap gestures. The user would then press and drag the thumbnails from their original positions and drop them in the viewing area. This would open a gene expression window combining the selected files (see Figure 5.10b). Both approaches are similar to those described when opening a single gene expression file (see Section 5.2.1).

It could also be possible to form combinations using a two-handed metaphorical gesture. With two open windows, a user would press on the title bars of each window and bring them close to each other until they intersect and merge into a combination of these two files. Using a similar gesture-based technique, a user could also remove files using one or more fingers. Ideally performed with two hands, the user would press on the title bar of a combination window with one hand and use the other hand to press on the name of a file to be removed from the window's side bar. The user would then drag the hands away from each other until the two windows, combination window and separate file window, were no longer combined (see Figure 5.10c).

Combining gene expression files using Add and Remove options are typically used in conventional setups and are easy to use in a touch UI where selection could be made with abstract tap gestures. This approach could also include additional options to combine or remove all files within a session or a combination.

Abstract discrete gestures, similar to those explored for opening a single gene expression file (see Section 5.2.1), could also considered but would require the user to perform multiple presses and taps to form a combination that would require the same effort as a menu option but none of its familiarity. Alternatively, using two hands windows could be made to intersect to form combinations. Despite the easily understandable metaphor for this technique, the large variable number of combinations that would typically be explored in an eQTL analysis requires more energy with this technique.

5.2.9 Match Significance across Files

Genetic variants collected from one gene expression file could be matched against another file to detect patterns and test combinations prior to formation. After selecting genetic variants or using the filtering technique (see Sections 5.2.4 and 5.2.7), genetic variants could be compared between files using menu controls. A user would first collect the filtered or selected genetic variants by tapping using one or more fingers on the Collect menu option. To compare the collected genetic variants against a file, the user would tap on the Compare menu option to match significance (see Figure 5.11).

It could also be possible to match significance using a two-handed metaphorical gesture. Similar to before, a user would first select individual genetic variants or filter a data set from one window. The user would then drag-and-drop the genetic variants onto another window. In either approach, intersecting pattern would be highlighted to be easily detectable. The overlapping of significant SNPs over gene expression files could imitate a lens approach that is commonly utilised in infovis,

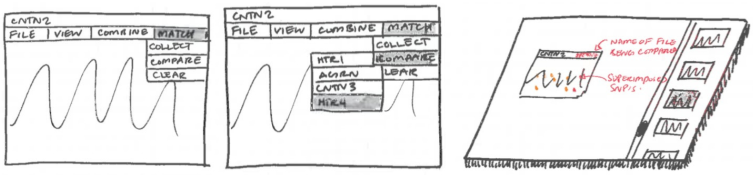


Figure 5.11: Match significance using a menu control. (left) Tap on the Match menu option. (middle) Tap on a subset to compare. (right) The subset is matched to the file's data set.

however since the overlap could span the entire genome the process of moving the lens would prove time consuming.

The options considered for matching significance across gene expression files attempt to balance conventional interactions and the potential energy required to perform the technique. The menu option would use the results of a filtered gene expression file to match significance with data points that meet the threshold. Menu options would be performed using abstract tap gestures common in touch UIs. An alternative approach would have users select and drag data points from one window to another in order to match significance. While the latter approach is more natural, the effort it would take to select multiple data points could prove time-consuming.

5.2.10 Desktop Idioms versus Gestures

The previous sections explore design options for a touch UI for infovis. All options considered are categorised as direct manipulation, but are explored separately as desktop idioms or gestures. An investigation of users' interactions with an interactive surface exposes the users' reliance on the desktop paradigm to guide interaction with a touch UI [56, 264]. The familiarity of WIMP interfaces introduces attractive options that can be utilised to explore eQTL results. Alternatively, gestures are designed to better reflect more natural and familiar everyday interactions that can lend themselves to exploring eQTL results.

Two types of gesture are explored in the design options for touch UI: one-handed and two-handed gestures. Nature and flow dimensions are also considered in the design options. Abstract gestures do not have a connection to symbolic, physical or metaphorical meaning and, therefore, the mapping of the gesture to an interaction is arbitrary. For example, a user may tap twice on a plot to zoom incrementally. Metaphors lend themselves to metaphorical gestures and thus give meaning to the connection between a gesture and the interaction. Expanding a window size by stretching the window from two opposite corners is an example of a metaphorical two-handed gesture [264]. Due to their stronger representation, metaphorical gestures are arguably more desirable than abstract gestures. However, the simplicity of an abstract gesture could prove beneficial. A gesture's flow can also be categorised as continuous or discrete, where continuous gestures are uninterrupted [264].

For opening, closing or hiding a gene expression file, a desktop paradigm (context menu) could be used with menu options to perform these actions. A user would then select an option using an abstract gesture. Another option to close or hide a window that uses a menu would anchor the menu bar and provide options to close or hide a window. Both alternatives are familiar and popularly used in WIMP interface, which is where their strength lies. Another strength of this technique is the ability to combine a number of options onto one control that is easily manipulated via tap gestures. This was also ideated for switching windows, forming combinations and matching significance across files. The strength of this approach is its familiarity and are often thought of when interacting with and interactive display [264], however it loses key benefits of gestures that better reflect everyday actions.

Metaphorical continuous gestures were also considered for opening and closing files via drag-and-drop. Similarly, hiding files could be achieved via a contracting metaphorical gesture. Other forms of metaphorical gesture utilise swiping to pan or scroll and pinch/spread to zoom. Both gestures are continuous and commonly used in smartphones and tablets. To form a combination, a metaphor of bringing files together and separating them via select and drag, was considered in the design options. Similarly, dragging a row from a table or a data point from a plot onto the viewing area could expose additional information about a genetic variant from external data sources. Translating and rotating gene expression windows was also achieved via metaphorical dragging or rotation and common one-handed or two-handed gestures. The metaphorical gestures considered for the touch UI are largely continuous, where the interaction is uninterrupted. This form of direct manipulation is relatively faster and adopts techniques from everyday behaviours that are easier to perceive, yet require more energy compared to abstract tap gestures that are used separately or within the context of a desktop idiom.

An abstract gesture's connection to its interaction is arbitrary but still comparable to other gestures that are metaphorical in nature. The explored design options considered abstract gestures for various eQTL tasks. A user could open a file by tapping on the file's thumbnail from the file explorer. Zooming could also be possible via a double tap gesture that zooms a plot incrementally. To switch windows, a user could adopt an abstract long press gesture to change from plot to table windows or vice versa. Despite their abstract nature, these gestures are largely easy to perform and could utilise natural or familiar interactions.

For the purpose of this research, we combine desktop idioms with familiar metaphorical and abstract gesture to balance the advantages and disadvantages of these approaches. The WIMP menu metaphor can easily combine functionalities that do not necessarily lend themselves to simple metaphorical gestures. For example, switching views via menu options instead of using a swipe gesture that could be better utilised to pan or scroll a plot or table window. Two-handed metaphorical gestures were considered for combining files and matching significance, however users were found to prefer less tiring one-handed gestures [264]. Drag-and-drop gestures were also found to cause more errors when comparing touch and mouse input (see [41, 67] and Section 4.4). Metaphorical gestures can better be adopted for simplified actions, such as filtering data sets by moving a threshold line in a plot window. Common gestures utilised for smartphones and tablets can also be adopted (e.g. zoom a plot using a pinch or spread gesture).

5.2.11 Touch UI Design Decisions

Various design options for a touch UI are considered for the development of an infovis touch UI for eQTL. The options considered are explored within two wider frames: desktop idioms and gestures. This was to balance the familiarity of conventional setups and the naturalness of everyday interactions that lend themselves to gestures. The touch interactions decided upon for the baseline touch UI are summarised in Table 5.4.

To open a gene expression file, a user would drag-and-drop a file's thumbnail from the file explorer to the viewing area. To close that file, a user would drag the open window to a graphical representation of a recycle bin. Both approaches adopt a press and drag gestures that is both natural and requires little energy. Well-established touch gestures for scrolling, panning and zooming would also be utilised for their familiarity. Metaphorical gestures would also be used to translate and rotate gene expression windows unimanually or bimanually. Genetic variants

eQTL Subtask	Touch Interaction
Open files	Drag-and-drop file from file explorer to the viewing area.
Hide files	Tap on Hide menu option.
Close files	Drag-and-drop file into graphical recycle bin.
Switch windows	Tap on Table or Plot menu options.
Scroll	Swipe in the opposite direction of intended scroll or use scrollbars.
Pan	Swipe in the opposite direction of intended pan.
Zoom	Pinch out/in to zoom in/out.
Select genetic variants	Tap on data point or row.
Expose information	Double tap on data point or row.
Deselect genetic variants	Tap on selected data point or row.
Open external sources	Tap on database's hyperlink.
Close external sources	Drag and drop window into graphical recycle bin.
Translate windows	Drag window on surface.
Rotate windows	One- or two-handed gestures to rotate window on surface.
Filter data	Select filtering from Threshold menu and rotate dial.
Add to combination	Select file from Group and Add menu options.
Remove from combination	Select file from Group and Remove menu options.
Match significance across files	Select collected subset from Threshold menu.

Table 5.4: The eQTL subtasks and the potential touch interactions for the baseline touch UI.

would be selected and deselected via tap gestures and additional information could be exposed by repeating the gesture more than once, i.e. double tap.

A number of the functionalities considered would be clustered into one control that is manipulated via tap gesture. This is to utilise the eQTL analysts' familiarity of desktop idioms and the ability to carry out a variable number of explorations. It would include the option to hide a gene expression file, switch views, combine files and match significance. In some instances these are used along with common gestures to access external data sources and match significance. Similarly table rows and plot data points would be filtered by combining menu selections with a metaphorical gesture to adjust the threshold.

5.3 TUI Design Options

This research envisions an infovis TUI with multiple tangible objects that are placed and manipulated on a planar surface. This type of tangible interface is classified as an interactive surface [246] and is an approach that is commonly utilised for many TUIs (see Section 2.2.2). The planar surface described in Section 5.1 is used with various tangible objects to explore the design options for the eQTL tasks. It is assumed that digital information is projected on the planar surface.

Three levels of data abstractions are manipulated during eQTL studies to identify areas or genetic variants of interest: combined gene expression files, independent gene expression files and subsets of files' data sets, i.e. SNP collections. Assuming an abstract representation, three different tangible objects may be used as containers for each level of abstraction. These will be known as the gene expression object, combined genes object and SNP objects throughout the design options discussion.

Cube-shaped objects are used extensively in prior work (e.g. [16, 30, 207, 236, 243]) due to peoples' intuitive understanding of how to manipulate them, as well as the flexibility with which cubes may be used. Cube-shaped objects are primarily illustrated here to represent gene expression files. Various shapes, sizes and colours are used to differentiate between the objects in the illustrations (see Table 5.5).

Meaning and purpose are often constructed from the physical representation of objects. This offers a rich opportunity to exploit various levels of an object's

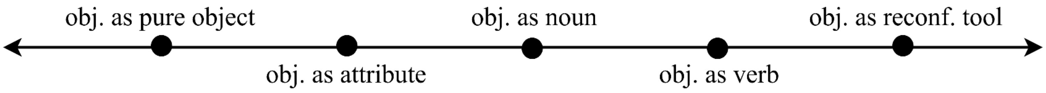


Figure 5.12: Tangible object's meaning continuum [248].

Object	Object type and description
⬚	Cube gene expression object
⊓	Planar surface
⬛	Docking box used to hide objects from view
⏢	Cone object used as a reconfigurable tool, i.e. menu object
🗔	Digital window
♻	Recycle bin
⊸	Slider object potentially used to pan, scroll and filter gene expression files
⬠	Prism object used to pan and scroll
🕹	Joystick to pan a visualisation
◓	Dial object used to filter a gene expression data set
◓	Dial object used to zoom a plot
🔍	Magnifying lens mainly used for navigational zoom
✎	Stylus for finer selections
⛁	Cylinder object mainly used to represent an external source
⬚	Cube combined genes object

Table 5.5: Tangible objects used to describe the TUI's design options.

specificity that may be utilised for tangible interaction. Underkoffler and Ishii [248] classify the meaning of tangible objects used on an interactive surface on a continuum (see Figure 5.12). Noun objects, lying in the centre of the axis, are physical representations of their digital counterparts. As the classification moves to the right of the continuum, objects become more generic and abstract. Verb objects are manipulated to alter the digital representation in a way that is not related to their physical representation. Further along the continuum, reconfigurable tool objects are completely abstracted from the physical presentation. To the left of the centre of the axis, objects are stripped of what can be done with them. Attribute objects only consider one single attribute of an object when manipulating digital representations. Further stripping the object results in a pure object where its existence is the only representation expected. Table 5.6 classifies the objects

eQTL subtask	Object classification			
	Attribute	Noun	Verb	Reconf. tool
Open files			●	
Hide files		●	●	●
Close files		●	●	●
Switch windows	●		●	●
Scroll	●		●	
Pan	●		●	
Zoom	●	●	●	
Highlight genetic variants				●
Expose information				●
De-select genetic variants				●
Open external sources	●		●	●
Close external sources	●			●
Translate windows		●		
Rotate windows		●		
Filter data			●	
Add to combination			●	●
Remove from combination			●	●
Match significance across files			●	●

Table 5.6: The various object's meaning classifications [248] considered for the TUI design options.

described in the following sections according to their position on the object's meaning continuum.

The TAC paradigm provides an understanding of the structure and functionalities of TUIs [212]. It identifies five components for describing a TUI: pyfo, token, constraint, variable and TAC. A pyfo is a physical object that takes part in a TUI and can either be a token or a constraint. A token is a graspable pyfo that is coupled with digital information or a computational function and whose behaviour is limited by another pyfo (known as a constraint). The term variable is used to describe digital information or a computation function in an application, which can either be represented by a token or semantically defined. A TAC defines the

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
1.1.a			Open file	Place object on thumbnail
1.1.b			Open file	Drag thumbnail to viewing area
1.2.a			Hide file	Place object in docking array
1.2.b			Hide file	Rotate object and place it on its hide side
1.2.c			Hide file	Place object on window, rotate to navigate selections, and tap object to select
1.3.a			Close file	Place object in the recycle bin to clear its content
1.3.b			Close file	Rotate object and place it on its close side
1.3.c			Close file	Place object on window, rotate to navigate selections, and tap object to select

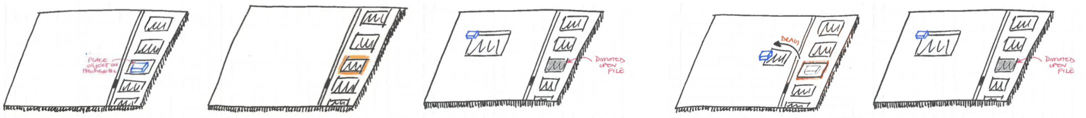
Table 5.7: Design options for opening, hiding and closing a gene expression file in a TUI using the TAC paradigm.

relationship between a token and its variable with one or more constraints and is represented with a number. In the following sections, we use the TAC paradigm to explore the design options for each eQTL task as well as illustrating these techniques when needed.

5.3.1 Open/Close Files

Statistically analysed gene expression files could be loaded and viewed as thumbnails in the file explorer. A user could open a file, close or hide it from view. In the TUI the gene expression files could be dynamically bound [244] to their objects so that they may be defined by the user of the system.

A user could open a gene expression file using a tap like gesture with the gene expression or combined genes object. The user would tap or place the object on



(a) Open a file using a tap gesture with an object. (left) Place an object on a file's thumbnail. (middle) The thumbnail's borders are highlighted in response. (right) Place the object on the viewing area to open the file's window.

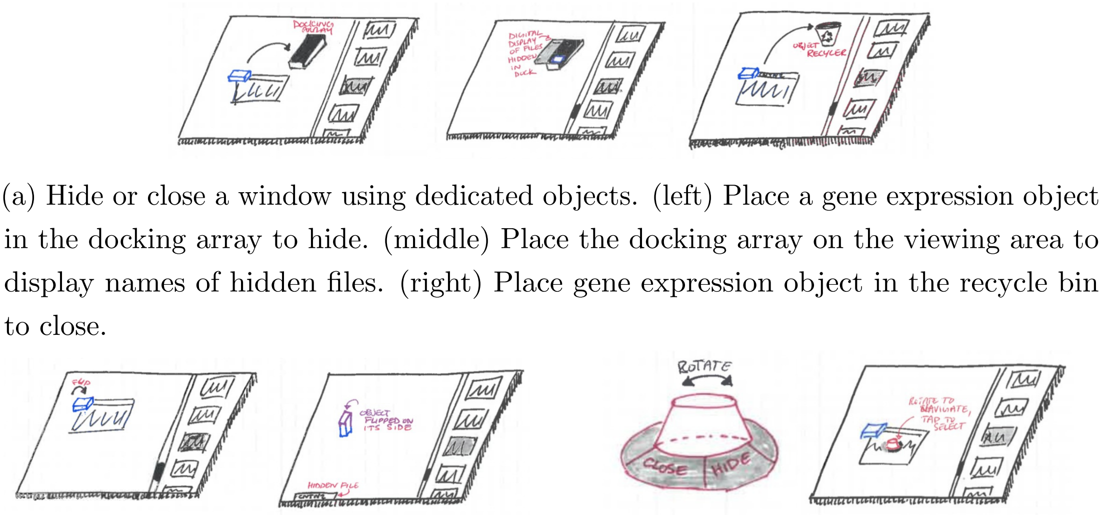
(b) Open a file using object drag. (left) Place an object on a file's thumbnail until it responds. (right) Drag the object to the viewing area to open the window.

Figure 5.13: Design options for opening a gene expression file in a TUI.

a file's thumbnail. The thumbnail would respond to the interaction, e.g. with a highlighted frame. The object would now be a container for the gene expression file. The user would place the object on the viewing area to open its window (see Figure 5.13a; TAC 1.1.a in Table 5.7). Alternatively, a drag-and-drop approach could be used to open a gene expression file. First, a user would place a gene or combined genes object on a file's thumbnail until it responds to the contact. The user would then drag the object onto the viewing area and as soon as the dragging action is complete the file's window is opened (see Figure 5.13b; TAC 1.1.b in Table 5.7). Once a gene expression file is opened, its thumbnail would be dimmed to disable any further interaction with it.

A gene expression object is envisioned as a container for the gene expression file's digital content. The act of coupling the digital and physical content would either have the user places a gene expression object on a thumbnail or drags the object onto the viewing area. In either case, contact would have to made between the thumbnail and the gene expression object. Placing the objects on the thumbnail and for the thumbnail to react better reflects the illusion of coupling, whereby dragging the object might reflect a transient object that does not necessarily maintain its one-to-one mapping with the digital content.

There are various ways to hide or close a gene expression's window depending on the specificity of the object being utilised. Dedicated objects, i.e. noun objects, could be used to physically hide or close a window (see Figure 5.14a). Using a docking array, a user would be able to hide a window from view by placing the gene expression's object inside of it (TAC 1.2.a in Table 5.7), which would hide


DOCKING ARRAY
DIGITAL DISPLAY OF FILES HIDDEN IN DOCK
OBJECT RECYCLER
FLIP
OBJECT FLIPPED ON ITS SIDE
HIDDEN FILE
ROTATE
CLOSE
HIDE
ROTATE TO NAVIGATE, TAP TO SELECT

(a) Hide or close a window using dedicated objects. (left) Place a gene expression object in the docking array to hide. (middle) Place the docking array on the viewing area to display names of hidden files. (right) Place gene expression object in the recycle bin to close.

(b) Hide or close a window using the object's many facets. (left) Rotate the gene expression object and place on its hide or close side. (right) The gene expression object is placed on its side to hide its window.

(c) Hide or close a file using a menu object. (left) A menu object with a series of options digitally circumventing it. (right) Place the menu object on a window and rotate to hide or close file.

Figure 5.14: Design options for hiding or closing a gene expression file in a TUI.

the window from view. When the docking array is placed in the viewing area, the names of the hidden files would be displayed digitally. Similarly, an object could be closed by utilising a physical recycle bin (TAC 1.3.a in Table 5.7). The user would place the gene expression object inside the recycle bin to unbind it from the file. The unbinding would also be reflected in the dimness of the file's thumbnail.

The various facets of a gene expression object could also be used to activate various options, such as hide and close (see Figure 5.14b). A user would flip the object and place it on its dedicated hide or close side to either hide or close a file's window (TAC 1.2.b and 1.3.b in Table 5.7). An alternative approach could use a generic menu object to navigate hide and close options (as well as other options discussed in later sections). A user would place the menu object on a gene expression's window, which displays a digital series of options circumventing the menu object. The user would rotate the menu object to navigate the options and tap on the top edge of the object to select an option (see Figure 5.14c; TAC 1.2.c and 1.3.c in Table 5.7).

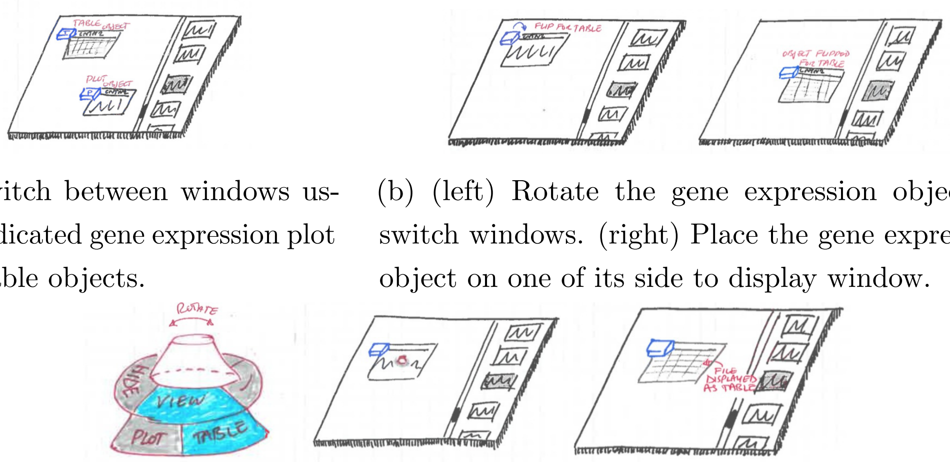
The closing and hiding of gene expression objects and their files considered noun, verb, and a reconfigurable tool. Noun objects operate as they would in the real world and in the case of closing or hiding a file, a user would physically place an object in a recycle bin to close and recycle the object or a box to hide objects and their files from view. Despite their strong specificity, their physical form would potentially clutter the space. Alternatively, the gene expression object many facets could be utilised by flipping the cube object along it sides to either hide or close a file. This approach maintains physical meaning within the contextual reference that would be provided by the eQTL application. A final approach considered using a generic object that is fully abstracted from any real-world representation. Similar to a mouse, the object would be used to make a selection by navigating a series of option. This approach strips the object of any meaning and essentially imitates a WIMP-style interface.

5.3.2 Switch Windows

When interpreting eQTL analysis results, users would typically switch between table and plot windows to aid the interpretation process. To switch between table and plot windows, various techniques could be used depending on the number and specificity of the objects. Windows could be switched using dedicated table and plot objects. This assumes that while one of the objects would be

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
2.a	P T	⊓	Switch windows	Place object on surface
2.b	□	⊓	Switch windows	Rotate object and place it on one its sides
2.c	□	▭	Switch windows	Place object on window, rotate to navigate selections, and tap object to select

Table 5.8: Design options for switching between a gene expression file's windows in a TUI using the TAC paradigm.



(a) Switch between windows using dedicated gene expression plot and table objects.

(b) (left) Rotate the gene expression object to switch windows. (right) Place the gene expression object on one of its side to display window.

(c) Switch between windows using a generic menu object. (left) A menu object with a series of options digitally circumventing it. (middle) Place the menu object on a window and rotate to switch windows. (right) The window is changed from plot to table.

Figure 5.15: Design options for switching between a gene expression file's windows in a TUI.

dynamically bound, the other should be statically bound by the developer to have both objects working in tandem. A user would place either the plot, table or both objects on the viewing area to display the corresponding window (see Figure 5.15a; TAC 2.a in Table 5.8).

Alternatively, the various faces of a gene expression object could be used to activate a number of different windows. A user would rotate the object and place it on its dedicated plot or table side to display and switch between windows (see Figure 5.15b; TAC 2.b in Table 5.8). A generic menu object could also be used to switch between windows. Similar to the technique described for closing or hiding a file (see Section 5.3.1), the user would place the menu object on the file's window, then rotate to navigate and tap to make a selection (see Figure 5.15c; TAC 2.c in Table 5.8).

An option to view table and plot view simultaneously could also be possible. A dedicated object for simultaneous viewing could be used. Alternatively, a user would place a table and plot window objects to view them side to side, which would give the user more spatial options. Another design option could present the simultaneous view as face on a gene expression object. The user would activate that view to display the windows at the same time. a generic menu object could also be used.

Three object classifications are considered for switching between table and plot view: attribute, verb, reconfigurable tool. Due to the abstract nature of the views, noun objects would not be considered. Using an attribute object to switch between views strips the objects of any meaning expect its position to display a plot or table window. It would also require the customisation of objects to represent each potential views, as well as supporting dynamic coupling to maintain concurrency. Similar to hiding or closing a file (see Section 5.3.1) a verb object could help maintain an object's contextual meaning. The use of a reconfigurable object abstracts the object even more, imitating the actions of a mouse (as seen in Section 5.3.1).

5.3.3 Scroll, Pan and Zoom

Plots and tables could interactively be manipulated to expose more information that is not directly shown. This could be achieved by scrolling, panning and zooming tables and plots. Dedicated objects for panning plots and scrolling tables could be used. To navigate a table window, a user could use a slider object. The user would place the object on the surface of the table window either horizontally or vertically. To scroll, the user would navigate the slider's handle in the opposite direction of the intended scroll (TAC 3.1.a in Table 5.9). To pan a plot window, a user could use a joystick to pan in joint directions. The user would place the joystick on the plot window and navigate its handle to pan the plot's view (TAC 3.2.a in Table 5.9).

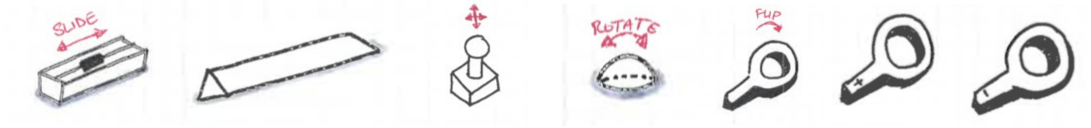
A generic object could also be used to either pan or scroll a table, e.g. a prism object with multiple contact points. A user would place the object on the window and then drag in the opposite direction of the intended pan or scroll (TAC 3.1.b and 3.2.b in Table 5.9). Figure 5.16a illustrates the three objects, two verb object and one attribute object, that could be used for the purpose of panning a plot or scrolling a table.

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
3.1.a	⊸	🗔	Scroll table	Place object on window and manipulate slider handle
3.1.b	▱	🗔	Scroll table	Place object and slide against window
3.2.a	🕹	🗔	Pan plot	Place object on window and navigate joystick's handle
3.2.b	▱	🗔	Pan plot	Place object and slide against window
3.3.a	◒	🗔	Zoom plot	Place object on window and rotate dial
3.3.b	🔍 🔍	🗔	Zoom plot	Place object on window
3.3.c	🔍	🗔	Zoom plot	Rotate object and place on window

Table 5.9: Design options for scrolling, panning and zooming a gene expression file's table or plot windows in a TUI using the TAC paradigm.

In a plot window, views could be zoomed using various techniques depending on the number and specificity of the objects being used (see Figure 5.16b). A user could zoom a plot using a dial object. To zoom, the user would place the dial object on the window and turn the dial clockwise or counter-clockwise to zoom in or out (TAC 3.3.a in Table 5.9). Alternatively, two dedicated magnifier objects could be used to zoom the plot's view. A user would place either the zoom in or zoom out magnifier object on the surface of the plot, which would zoom the plot using preset increments (TAC 3.3.b in Table 5.9). To reduce clutter, a single magnifier object could be used to zoom in and out of a plot. The user would place the magnifying object on one of its two sides on a plot window to zoom (TAC 3.3.c in Table 5.9).

Within the context of navigating a plot or table window, verb object are considered for scrolling, panning and zooming. Relatively specific objects (e.g. slider and joystick) could be used to pan or scroll a window and as physical representation of graphical facsimiles provide familiar and intuitive manipulation. An abstract object could also be stripped of meaning excluding its velocity to pan or scroll a window or its rotation to zoom in/out. However, this would raise the issue of clutter since potentially relevant meaning is stripped of the object.



(a) Objects to pan plots and scroll tables. (left) A slider to manipulate and scroll a table or pan plot. (middle) An abstract object to brush window to pan or scroll. (right) A joystick to pan a plot.

(b) Objects to zoom a plot. (left) A ial object to rotate and zoom. (middle) One magnifier object to zoom. (right) Two magnifier objects, one to zoom in and the other to zoom out.

Figure 5.16: Design options for scrolling, panning and zooming a gene expression file's table or plot window in a TUI.

For zooming in and out of plots, a highly specific noun object would be used. Despite the risk of clutter, the strong specificity support intuitive manipulation of the objects.

5.3.4 Select Genetic Variants

Hundreds of thousands of genetic variants should be plotted against their significance in each file, and a finer grade of selection would be required to highlight their importance or to expose additional information. A stylus-shaped object could be dedicated for the selection of genetic variants. A user would use the stylus to tap once to select a row or data point or twice to display additional information (TAC 4.1 and 4.2 in Table 5.10). To deselect a genetic variant, the user would tap on the data point or row once again (TAC 4.3 in Table 5.10). Due to the abstract nature or the data set and its size, finer selections are required and would be best supported with a familiar stylus selections.

5.3.5 Access External Sources

Access to external data sources aids users when exploring eQTL results and facilitates the process of interpreting and understanding the results. A number of representative objects could be dedicated to relevant external data sources, i.e. attribute objects (see Figure 5.17a; TAC 5.1.a in Table 5.11). A user would access an external data source, e.g. dbSNP, by first selecting a genetic variant (see

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
4.1	✎	🗔	Select genetic variant	Tap on genetic variant
4.2	✎	🗔	Expose information	Double tap on genetic variant
4.3	✎	🗔	Deselect genetic variant	Tap on genetic variant

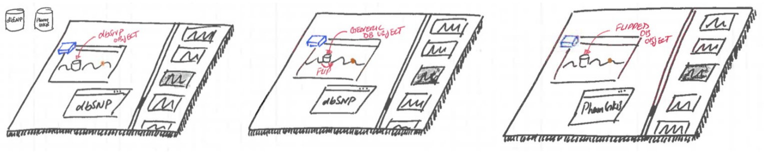
Table 5.10: Design options for selecting genetic variants in a TUI using the TAC paradigm.

Section 5.3.4). The user would then place the dbSNP object on the file's window. This would display a window with information about the selected genetic variant from the external source.

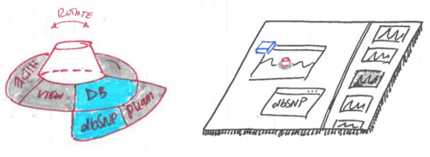
Alternatively, instead of dedicating multiple objects, a single object could be used and its sides utilised for access (TAC 5.1.b in Table 5.11). A user would be able to retrieve information from a particular data source by placing the external data source object on the desired data source side (see Figure 5.17a). A generic menu object could also be used to access an external data source. A user would place the menu object on a file's window, then rotate to navigate and tap to make

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
5.1.a	⛁ dbSNP ⛁ PharmGKB	🗔	Open external source	Place object on window
5.1.b	⛁	🗔	Open external source	Place object on window
5.1.c	⏢	🗔	Open external source	Place object on window, rotate to navigate selections, and tap object to select
5.2	⛁	🗔	Close external source	Remove object from window

Table 5.11: Design options for accessing external data sources in a TUI using the TAC paradigm.



(a) Access external source using attribute objects. (left) Place an external object on a window to open source. (middle) Rotate the external source object to its open side. (right) Place the external source object on one of its sides to open source.



(b) Access an external source using a generic menu object. (left) A menu object with a series of options digitally circumventing it. (right) Place the menu object on a window and rotate to navigate and tap to select an external source to open.

Figure 5.17: Design options for accessing external data sources in a TUI.

a selection (see Figure 5.17b; TAC 5.1.c in Table 5.11). An external source's window could be closed by removing the object from the file's window (TAC 5.2 in Table 5.11).

For the purpose of accessing external data sources attribute, verb and reconfigurable objects were considered. The nature of an attribute object would strip it of meaning and risk clutter by utilising one object per data source. By using a verb object, multiple source would be combined to form a data source object with multiple access options. This not only reduces clutter but also utilises the many facets of the objects and supports expansion. Alternatively, a generic object could potentially be used. While it could combine multiple functionalities discussed in the previous sections, it similarity to post-WIMP interfaces and interaction defies the vision of TUIs.

5.3.6 Organise Windows

Multiple files could be viewed at the same time and rearranged to help identify patterns and ease interpretation. In a tangible interface, a user could move a

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
6.1	⧉	⊓	Translate window	Move object on surface
6.2	⧉	⊓	Rotate window	Rotate object on surface

Table 5.12: Design options for organising windows in a TUI using the TAC paradigm.

window by moving its object around the viewing area (TAC 6.1 in Table 5.12). A user would either pick up the object and then place it in another position or drag the object across the screen to move. The former would result in the digital representation being closed and then reopened once the object is replaced in another position. The latter would move the digital representation to follow the motion of its object. A user could also reorient a window by rotating its object around its z-axis (TAC 6.2 in Table 5.12). As a physical container for a digital files, a gene expression object supports a noun meaning where it is operated by the user as it would in the real world to translate and rotate its content.

5.3.7 Filter Data

When working with large data sets, users would often find filtering essential to work with a subset of the data set (e.g. only view genetic variants that meet a significance threshold). Specific objects could be dedicated to filtering data sets. Using a slider object, a user would be able to adjust the significance threshold

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
7.a	⊸	▭	Filter data	Place object on window and manipulate slider handle
7.b	⬭	▭	Filter data	Place object on window and rotate dial

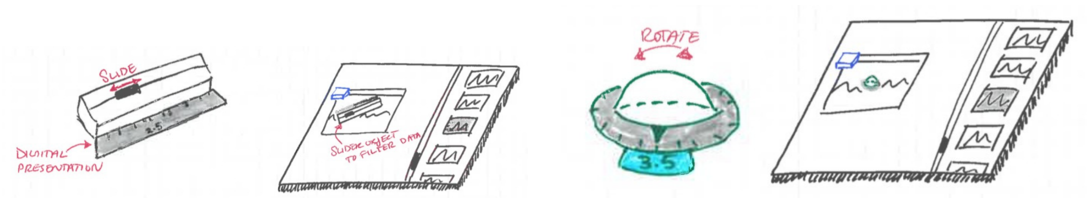
Table 5.13: Design options for filtering a gene expression's data set in a TUI using the TAC paradigm.

by placing it on a file's window. This would display the threshold range and increments digitally (each gene expression file should have its own significance range given its maximum and minimum significance). To adjust the threshold, the user would move the slider's handle (see Figure 5.18a; TAC 7.a in Table 5.13).

A dial object could also be used to filter the data. A user would place the dial object on a file's window and would react by showing the threshold range and filtering increments around the dial. The user would then rotate the dial clockwise or counter-clockwise to adjust the threshold (see Figure 5.18b; TAC 7.b in Table 5.13). A filtered plot window would cause the non-significant data points to be dimmed, while in a table window rows that do not meet the significance threshold would be collapsed from view.

An alternative approach could adopt the concept of a semi-permeable filter used in Kinetica [199]. A physical object would be dedicated to perform as a magnet that attracts data points that meet a threshold requirement when placed on a gene expression window. The threshold requirement for the object would be set when the object is placed on the viewing area and navigated via touch. The object would be categorised as a verb object and imitates the performance of the magnet filtering objects used in Vispol [147] to filter case-based data sets.

Within the context of filtering a data set in a plot or table, verb objects such as a slider or a dial would be used to work with a subset of the data set. Similar to scrolling a table (see Section 5.3.3) a user could potentially manipulate a physical slider to adjust a threshold criterion. Another familiar real world object for adjusting values would be a dial that could be rotated to adjust a threshold.



(a) (left) Filter data using a slider. (right) Place the slider on a window and navigate handle to adjust threshold and filter.

(b) (left) Filter data using a dial. (right) Place the dial on a window and rotate to adjust threshold and filter.

Figure 5.18: Design options for filtering a gene expression's data set in a TUI.

Of the two choices, the use of a dial would potentially avoid confusion for when a slider is also utilised to scroll a table. Other alternative, such as an attribute or reconfigurable objects, were not considered as objects with higher specificity were desired to manipulate data sets. The use of magnet object presents an attractive options, but the two step set up of the threshold value would require more energy compared to using a dial object.

5.3.8 Combine Files

Patterns shared between gene expression files would typically be detected after viewing each file separately. Combining gene expression files would allow for the identification and exploration of patterns from multiple files at the same time. Combinations could be formed using a generic menu object. A user would place the menu object on a window to display options to add or remove to/from a combination. The user would then rotate to navigate and tap to make a selection. A combination window would superimpose plots and merge tables, while also displaying a side bar with the names of the files in the combination (see Figure 5.19a; TAC 8.1.a and 8.2.a in Table 5.14).

Using a similar approach used to open a gene expression file (see Section 5.3.1), a user could form a combination using a combined genes object. The user would place or tap a combination object on multiple files' thumbnails in the file explorer. The combined genes objects is now a container for multiple files. The user would open the combination window by placing the combined genes object on the viewing area (see Figure 5.19b; TAC 8.1.b in Table 5.14).

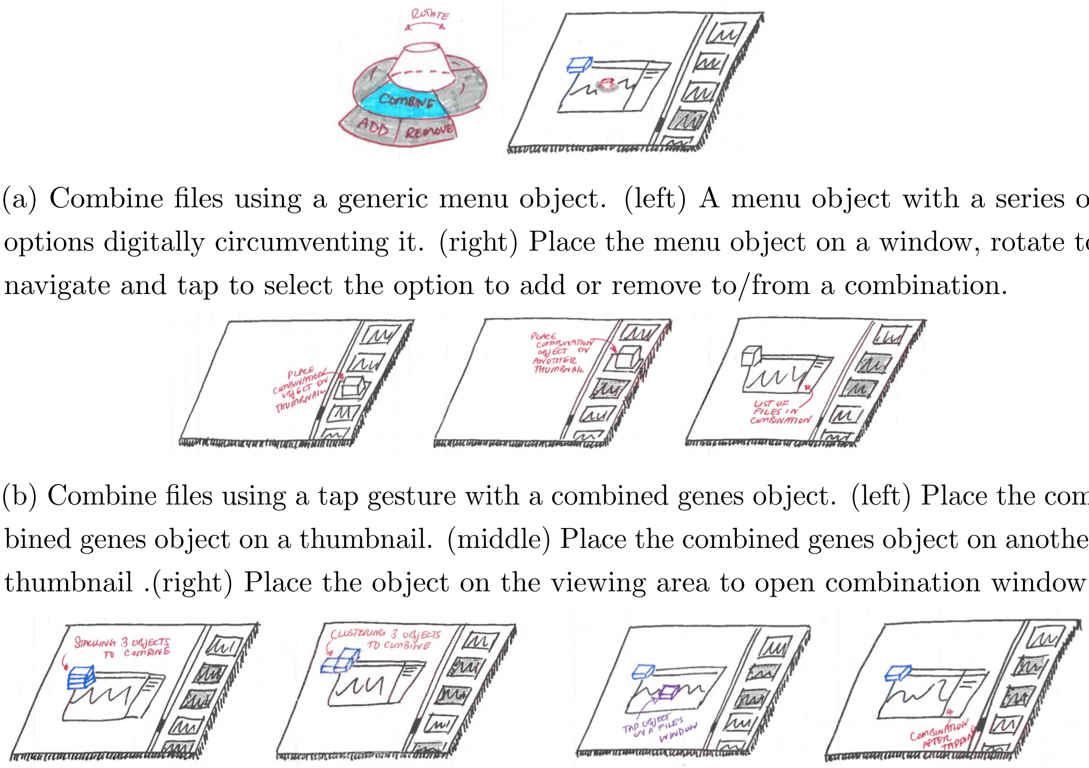
Gene expression objects representing individual files could also be used to form combinations. A user could cluster gene expression objects together to form a group. The objects' windows would respond to the proximity of the objects and merge to form a combination window. Alternatively, a user could stack gene expression objects one on top of the other to form a combination (see Figure 5.19c; TAC 8.1.c and 8.1.d in Table 5.14). A file would be removed from a combination window by physically removing the gene expression object from the cluster or stack (see TAC 8.2.b and 8.2.c in Table 5.14).

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
8.1.a	⏢	🗔	Add to combination	Place object on window, rotate to navigate selections, and tap object to select
8.1.b	❒	⊓	Add to combination	Place combination object on multiple thumbnails
8.1.c	❒	⊓ ❒	Add to combination	Stack object one on top of the other
8.1.d	❒	⊓ ❒	Add to combination	Cluster objects close to each other
8.1.e	❒	🗔	Add to combination	Tap or place object on another window
8.2.a	⏢	🗔	Remove from combination	Place object on window, rotate to navigate selections, and tap object to select
8.2.b	❒	⊓ ❒	Remove from combination	Remove object from stack
8.2.c	❒	⊓ ❒	Remove from combination	Remove or move object away from cluster
8.2.d	❒	🗔	Remove from combination	Tap or place object on another window

Table 5.14: Design options for combining gene expression files in a TUI using the TAC paradigm.

Another approach could utilise a tapping gesture with the objects to form a combination. In an open file's window, a user would tap another file's object onto the window's surface. This would merge the tapped object's file onto the displayed window. To remove a file from the combination, the user would similarly tap again on the surface of the displayed file's window (see Figure 5.19d; TAC 8.1.e and 8.2.d in Table 5.14).

Various options are considered for the purpose of combining gene expression files and explore significance across files. As abstract containers for gene expression


ROTATE
COMBINE
ADD
REMOVE
PLACE COMBINATION OBJECT ON THUMBNAIL
PLACE COMBINATION OBJECT ON ANOTHER THUMBNAIL
LIST OF FILES IN COMBINATION
STACKING 3 OBJECTS TO COMBINE
CLUSTERING 3 OBJECTS TO COMBINE
TAP OBJECT ON A FILE'S WINDOW
COMBINATION AFTER TAPPING

(a) Combine files using a generic menu object. (left) A menu object with a series of options digitally circumventing it. (right) Place the menu object on a window, rotate to navigate and tap to select the option to add or remove to/from a combination.

(b) Combine files using a tap gesture with a combined genes object. (left) Place the combined genes object on a thumbnail. (middle) Place the combined genes object on another thumbnail .(right) Place the object on the viewing area to open combination window.

(c) (left) Combine files by stacking gene expression objects or (right) clustering objects.

(d) (left) Tap a gene expression object on a window. (right) The two files are combined in the window.

Figure 5.19: Design options for combining gene expression files in a TUI.

files, gene expression objects lend themselves to being used as verb objects. To form a combination, a user would either stack or cluster gene expression objects together. This behaviour is commonly utilised and could be easily perceived as a metaphor for a group or combination. Alternative options include using a generic mouse-like object or another verb object where combination could potentially be tapped. Another options considered utilised an additional combination objects that is specifically used for the purpose of combining gene expression files. This would require introducing an additional abstract object and a set of interactions specific to the object, which could potentially increase the application's complexity given simpler alternatives.

5.3.9 Match Significance across Files

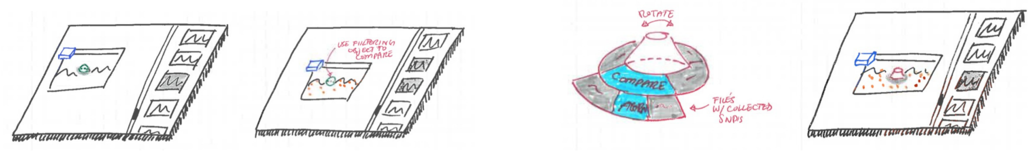
To detect patterns and test combinations, genetic variants could be collected from one gene expression file and compared against another. The dial object described in Section 5.3.7 could be used to collect and match significance. A user would first collect genetic variants from one window by filtering a data set and collecting a subset. The user would then place the same dial object on another window. This would cause intersecting patterns to be highlighted (see Figure 5.20a; TAC 9.a in Table 5.15).

Alternatively, a generic menu object could be used to collect and compare variants across gene expression files. A user would first place the menu object and navigate its options to collect genetic variants by filtering or selecting a subset (see Sections 5.3.4 and 5.3.7). The user would then place the menu object on another file's menu, rotate to navigate options and tap to compare significance (see Figure 5.20b; TAC 9.b in Table 5.15).

Two options are considered for matching significance across gene expression files to balance object specificity and function. Reconfigurable tools have been discussed in previous sections and particularly its resemblance to mouses used in traditional setups. Alternatively, the use of a verb object could overcome some of these problem as the object is given meaning within the context of the application and objects. In this case, the object utilised for filtering a data set would simultaneously store the filtered subset to be matched for significance against

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
9.a	⬭	🗔	Match significance	Place object on another window
9.b	⏢	🗔	Match significance	Place object on window, rotate to navigate selections, and tap object toselect

Table 5.15: Design options for matching genetic variants' significance across gene expression files in a TUI using the TAC paradigm.



(a) Match significance using a dial object. (left) Place the dial object on a window, filter data by rotating the object and collect genetic variants. (right) Place the dial object on another window to match significance.

(b) Match significance using a menu object. (left) A menu object with a series of options digitally circumventing it. (right) Place the menu object on a window, rotate to navigate and tap to select option to match.

Figure 5.20: Design options for matching genetic variants' significance across gene expression files in a TUI.

other files. This would reduce the number of objects employed with the system, and also would maintain a link between the data collected and those matched.

5.3.10 Objects' Specificity and Interpretation

The previous sections describe design options that interact with objects at various levels of specificity (see Table 5.6). The user's perception of function from an object is dependent on its physical appearance and their personal experience with familiar objects. The continuum of object meaning classifies interactive surface objects along two directions that divert from a centre representing a real-world object [250] (see Figure 5.12). The importance of exploring different levels of abstractness lies in its effect on the interpretation of interaction, functionality and space clutter. On one hand, the stronger the specificity the more difficult it is to combine functionalities in a single object to reduce clutter in a fully functional tangible system. General objects, on the other hand, lack meaning, making it more difficult to fathom function from their abstract form [86].

To the left of the continuum of object meaning centre, objects are stripped of their real-world meaning except for one of their properties. Therefore, the tangible system only recognises a single property of the object. Attribute objects have been used in previous TUIs for infovis. In previous work, attribute objects were developed to save a system state in a police case TUI [147]. A user would place a uniquely identifiable object on the interactive surface in order to save a state. Merely its existence on the surface is of significance to the system. Two layout tangibles were also used in the system to switch between layout. A user would place a layout tangible on the surface to change the display, but neither its position, orientation nor shape are of any significance to the system.

Attribute objects were one of the options considered for switching between plot and table windows. Dedicated plot and table objects, imagined as abstract cube objects or possibly 3D printed objects to provide a sculpted representation, only need to be placed on the surface in order for a window view to be displayed. The object's position is the only attribute that is considered by the system. This was also the case with dedicated external sources that could be represented with objects that are stripped of all meaning except for their position. Similarly, panning or scrolling a window considered the velocity of the ideated object to perform an intended pan or scroll. While objects as attributes support affordance, yet not as strongly as noun objects, they risk clutter due to the fact that the objects may be stripped of potentially useful properties. For example, a cube object could be flipped, rotated and moved to afford useful functionalities; however as an attribute object, these dynamic properties are ignored.

Further to the left of the centre of the object meanings continuum lie objects as pure objects. Pure objects are stripped of any intrinsic meaning and, therefore, their only function is to exist as objects, which are not necessarily uniquely identifiable. This approach falls at risk of stripping any form of affordance and thus meaning from an object and was not considered in the research's exploration of the design options.

Noun objects operate as they do in the real world. For example, the use of a cube object to translate and rotate a digital representation caters to the object's familiarity and potential functionalities. This was particularly evident when using a metaphor of stacking or clustering to form groups or combinations. However, it can be argued that these types of objects lie between noun and verb objects. This is due to the object's abstract nature and the object's placement within the context of eQTL analysis. Other examples of noun objects in the TUI design options utilised box and bin metaphors to hide or close gene expression files. A lens object was also considered to zoom within a plot, where the lens imitates a real-world magnifier. The strength of noun objects lie in their strong specificity. They are rich with familiar information that guide users' actions to perform an intended functionality. However, abstract data lack a familiar physical form and may prove difficult to represent as a noun object. The strong specificity of the objects make them harder to reuse or recycle.

Under the category of physical interfaces, inFORM [232] is an example of a strongly specific TUI where information is physically presented in the real-world. inFORM represents bar charts physically and explores the interaction space for the physical interface. Urp also utilises noun objects that represent wireframe building and roads for an urban planning TUI [250]. Lens-based application (e.g. [55, 124, 134, 227], adopt the noun form of a lens to isolate or focus a subset of an infovis. However, in both cases, the interaction with the noun objects has gone beyond the familiar to expand the interaction space. This can arguably place the objects somewhere between noun and verb objects on the continuum.

To the right of the continuum lie objects that are classified as verb objects or reconfigurable tools. Verb objects are stripped of their real-world meanings and are placed within a contextual reference to extend their functionality. In the TUIs for infovis literature, verb objects are commonly utilised to interact with the abstract data sets. Urp's wind simulation and shadow clock tool are characterised as verb objects [250]. The active tangibles in the sparse tangible interface are used to explore genomic networks and are verb objects; this is because the abstract objects are manipulated in a familiar way within the context of a gene network, e.g. stacking objects to filter [9]. In a police case application, magnet objects are used to filter case-based data sets by attracting subsets that meet the object's criteria [147]. Other examples of verb objects are found when exploring architectural tourist spots in Venice [169] and using a ring metaphor to manipulate a visualisation of abstract data [55]. Generally, verb objects are abstract to extend functionality and allow for richer interaction without the clutter. However, their abstract nature lessens their affordance to the contextual task.

Verb objects were considered in the design options for the TUI for various eQTL tasks. A physical slider was considered within the context of a plot or table window to pan or scroll, respectively. Alternatively, a joystick could offer more degrees of freedom when panning a visualisation. These tools can arguably lie in between noun and verb classification of an object as they are proxies of objects that occupy the digital world. For example, a tangible slider is a physical representation of a scrollbar and meaning can easily be grasped from the object. The familiarity of cube objects presented an attractive option to exploit their manipulation for hiding or closing a file by merely flipping an object on a dedicated side. This action has previously been used in the literature (e.g. [169, 227]). To filter an eQTL data set, a dial object was considered as an example of a verb object that is rotated to navigate filtering options. This approach has been adopted in previous infovis TUI research (e.g. [40, 55, 169, 245]).

Further to the right of the continuum are reconfigurable tools that are fully abstracted from their real world representation. A popular example of such an object is the mouse. The design options described a menu object that, when placed on the surface, could provide options for switching windows, accessing external sources, forming combinations and matching significance. A user would rotate the object to navigate the options and make a selection by tapping on the object. This approach makes it easier to combine functionalities onto one object and, in effect, imitates a WIMP-style interface. It can be argued that the multi-faceted object used to explore architectural spots in Venice is a reconfigurable tool [55]; however, it is not as fully abstracted since the user may flip the object and place it on any of its facets.

5.3.11 TUI Design Decision

Various design option for a exclusively tangible UI are considered in this section for the eQTL infovis application. The options are explored along an object's meaning continuum (see Figure 5.12). Table 5.16 summaries the interactions considered for the exclusively tangible eQTL TUI.

Of the various types explored in the design options, verb objects are the likeliest contenders for an exclusively tangible infovis TUI. This is due to the fact that functionalities would be packed in a smaller number of objects, but not to the extreme (i.e. reconfigurable tools). The abstract nature of verb objects would also lend themselves to the nature of infovis data, without the risk of stripping the utilised object of functionality (i.e. attribute objects). Noun objects are an

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
1.1	❒	⊓	Open file	Place object on thumbnail
1.2	❒	⊓	Hide file	Rotate object and place it on its Hide side
1.3	❒	⊓	Close file	Rotate object and place it on its close side
2	❒	⊓	Switch windows	Rotate object and place it on surface on one of its sides
3.1	🕹	🗔	Pan plot	Place object on window and navigate joystick's handle
3.2	🕹	🗔	Pan plot	Place object on window and navigate joystick's handle
3.3	🔍	🗔	Zoom plot	Rotate object and place on window
4.1	✎	🗔	Select genetic variant	Tap on genetic variant
4.2	✎	🗔	Expose information	Double tap on genetic variant
4.3	✎	🗔	Deselect genetic variant	Tap on genetic variant
5.1	⌭	🗔	Open external source	Place object on window
5.2	⌭	🗔	Close external source	Remove object from window
6.1	❒	⊓	Translate window	Move object on surface
6.2	❒	⊓	Rotate window	Rotate object on surface
7	◠	🗔	Filter data	Place object on window and rotate dial
8.1	❒	🗔 ❒	Add to combination	Stack object one on top of the other
8.2	❒	🗔 ❒	Remove from combination	Remove object from stack
9	◠	🗔	Match significance	Place object on another window

Table 5.16: An exclusively tangible TUI's core elements and interactions described using the TAC paradigm.

attractive concept but are unrealistic for representing abstract large data sets but could prove useful for other forms of visualisations, e.g. scivis.

5.4 Hybrid TUI Design Options

Hybrid interfaces combining touch and tangible interaction are commonly seen in TUI literature (e.g. [55, 191, 204, 211]). Both interaction techniques are designed to exploit humans' motor skills and the cognitive models gained from interacting with real objects in the real world. With touch interaction, metaphorical digital representations are often utilised to suggest interactions. Tangible interactions are often carried out with verb objects, where the object's degrees of freedom and affordance are exploited. Both touch and tangible interaction promote bimanual interaction, space-multiplexing and parallel input. The balance between physical and digital representations is one of TUIs' greatest design challenges [126, 234, 244]. In this section, the design options described for the touch UI and TUI systems are combined to balance physical and digital representation. Table 5.17 summarises the hybrid TUI's core subtasks and interactions using the TAC paradigm.

5.4.1 Open/Close Files

A gene expression object is envisioned as a container for the gene expression file's digital content [126]. The action of opening a file is linked to the appearance of its digital content as the object is placed on the surface. This strong coupling maintains the one-to-one mapping between the object and its digital content, i.e. deactivating interaction with a gene expression file's thumbnails when linked to an object. The manipulations considered for linking the object with the digital content seems appropriate to reflect the illusion of the object as a container and overcomes the discontinuity of opening a file via touch (tap and double tap).

Design options for hiding or closing a file consider verb (sides of an object), noun (physical containers) and reconfigurable tool (menu object) objects (see Section 5.3.1). The strengths and limitations of the various objects were previously discussed (see Section 5.3.10). For the purpose of hiding a window from view, the affordance of the gene expression object's is utilised. This approach reduces clutter and introduces an implicit user-maintained mode [126]. In infovis TUI literature, an object's many facets has been utilised for various functionalities (e.g. [169, 227]). For example, a multi-faceted object was used to explore and filter architectural spots in Venice [169].

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
1.1	🧊	⊓	Open file	Place object on thumbnail
1.2	🧊	⊓	Hide file	Rotate object and place it on its hide side
1.3	🧊	⊓	Close file	Drag object to a graphical recycle bin
2	🧊	⊓	Switch windows	Rotate object and place it on surface on one of its sides
3.1	✋	🗔	Scroll table	Press with fingers and move
3.2	✋	🗔	Pan plot	Press with fingers and move
3.3	✋	🗔	Zoom plot	Pinch and spread
4.1	✋	⊓	Select genetic variant	Tap on genetic variant
4.2	✋	⊓	Expose information	Double tap on genetic variant
4.3	✋	⊓	Deselect genetic variant	Tap on genetic variant
5.1	✋	⊓	Open external source	Tap on external source
5.2	✋	⊓	Close external source	Tap on close
6.1	🧊	⊓	Translate window	Move object on surface
6.2	🧊	⊓	Rotate window	Rotate object on surface
7	◓	🗔	Filter data	Place object on window and rotate dial
8.1	🧊	🗔 🧊	Add to combination	Stack object one on top of the other
8.2	🧊	🗔 🧊	Remove from combination	Remove object from stack
9	◓	🗔	Match significance	Place object on another window

Table 5.17: Hybrid TUI's core elements and interactions described using the TAC paradigm.

A gene expression window could temporarily be closed by removing its object from the surface. To permanently close a window, a graphical representation of a recycle bin would be used in place of using an additional object. A gene expression object is placed on or dragged to the recycle bin to severe the link between the object and its gene expression file. The rationale behind this decision equates the semantic meaning acquired from a physical recycle bin and its digital equivalent. Therefore, the hybrid TUI relies on the digital convention to convey the message. Arguably, the graphical icon can be updated to reflect the recyclability of the object itself.

5.4.2 Switch Windows

Switching between gene expression windows was considered with the use of specific objects representing the various window options (plot or table) as well as utilising the affordance of the cube gene expression object to switch between windows. The use of one object to represent a file better reflects the one-to-one mapping between the object and its digital content. This approach also reduces clutter and the risk of losing physical objects. Therefore, utilising the facets of a gene expression object is deemed suitable for the limited number of window options in consideration.

Gestures, using one or more fingers, were examined to switch between windows in the touch UI. While our system provides a small subset of functionalities, the use of unique gestures introduces another level of complexity that may have an adverse effect on the physical container concept. Also the swiping gesture considered for switching windows is commonly used for panning on smartphones and tablets, and could possibly interfere with that simplified mental mode. While it is possible to disambiguate the effects of the swiping gesture based on context, this is not ideal for panning and switching windows since they occur within the gene expression window.

5.4.3 Scroll, Pan and Zoom

When manipulating gene expression windows, the user's focus is diverted from the gene expression object and towards its digital content. The possibility of introducing new objects, either strongly, contextually specific (noun or verb) or generic (reconfigurable tool), to manipulate digital content can cause distraction that shifts a user's focus. Scrolling, panning and zooming have established touch gestures that are commonly used with smartphones and tablets. Views are often shifted using brushing or swiping gestures in order to scroll or pan content. Zooming is typically performed using a pinch and spread gesture that can be performed using one or two hands. These common interactions are well-suited to navigate the digitally presented infovis naturally and easily.

5.4.4 Select Genetic Variants

The selection of genetic variants to highlight or expose additional information was considered using touch and a reconfigurable tool, i.e. stylus. Both approaches adopt the same techniques, tap to highlight and double tap to expose information. Voicing the same concerns discussed in Section 5.4.3, the use of touch is deemed more appropriate. The use of touch also reduces the effort it may take to grasp a stylus; an added step compared to the more natural touch technique. TUIs for infovis similarly adopt abstract tap gestures for selecting options or data points [9, 55, 111, 214].

5.4.5 Access External Sources

The design options considered for accessing an external data source link the interaction with the selection of genetic variants. A tangible external data source object (attribute or verb object) could situate the source's presence in the real world, but risk clutter. Alternative touch gestures utilise abstract tapping or metaphorical gestures. Tapping on a linked external source is familiar, but combined with a double tap to expose information results in three discrete gesture to access a data source. The alternative metaphorical drag-and-drop is an attractive option, however it falls at the risk of increasing the error rate for touch (see [41, 67] and Section 4.4). A more direct approach could combine the selection of a genetic variant and exposing additional information (see Section 5.4.4) with default access to an external data source. This reduces the number of taps required to access an external source and potentially lowers the risk of drag-and-drop error.

5.4.6 Organise Windows

With the various tangible objects acting as containers for data sets (separate gene expression files, combinations of files or a subset of a data set), organising windows spatially around a surface is strongly coupled to the placement and orientation of the object. This reinforces the illusion that the physical and digital worlds are connected. While touch gestures for translating and rotating windows are well-established and familiar, they dissociate the object from its digital content and negatively affects the concept of utilising the gene expression object as a container for the digital file.

5.4.7 Filter Data

The tangible options (slider and dial) examined for filtering a data set are both contextually specific to the action in hand (verb objects). Similar graphical approaches manipulated via touch were also considered (slider, drop down menu and text input). The same concerns mentioned in Sections 5.4.3 and 5.4.4 could arguably stand for filtering and collecting data. However, the situatedness of a tangible object in the physical world introduces a degree of importance to the interaction and the filtered subset. Essentially allowing the subset to exist outside of its parent file.

5.4.8 Combine Files

Physical building blocks are intuitively manipulated using one or two hands to form constructions. Building blocks have regularly been utilised with several TUIs (e.g [16, 30, 70]) using different mechanisms ([6, 64]). Clustering and stacking objects are popular approaches to combine filters or aggregate files in infovis TUIs (e.g. [9, 108, 111, 133]). Compared to the other approaches considered in the touch UI and TUI design options (see Sections 5.2.8 and 5.3.8), clustering and stacking objects seems to be the best at representing the link between the physical manipulation and the combined digital content.

Forming combinations by clustering or stacking objects introduce limitations that should be taken into consideration. Clustering objects could monopolise the surface space as the combination grows larger in size, not only obscuring its own digital content but also that of other objects. Stacking objects instead overcomes the space issues, however it introduces another issue relating to the stack's height. This limits the combination size and could accidentally be knocked over (unless stacks are physically linked, e.g. lego). Considering combinations limited in size, stacking is the more attractive option as it better represents the digital behaviour of the objects' content (superimposing data sets over each other, i.e. stacking the data sets digitally).

5.4.9 Match Significance across Files

A number of touch gestures were considered for the touch UI (desktop idiom and metaphorical gestures) using the subset extracted when filtering the data set. The desktop idiom approach requires several discrete abstract gestures to match significance, and the metaphorical gesture necessities the use of two hands to complete. Either approach introduces a level of complexity that could be avoided with a tangible object that could be manipulated eyes-free and while only using one hand.

The object used for filtering and storing a subset of a gene expression file (see Section 5.4.7) can ideally be used to match its significance against another file. The object would be utilised as a verb object and placed on another window to highlights matches. This approaches strengths the concept of using that object as a container for the subsets and can potentially promote epistemic actions and eyes-free interaction.

5.4.10 Hybrid TUI Design Decisions

This section explored the balance between physical and digital representation by combining the design options discussed in the previous sections and summarised in Tables5.4 and 5.16. The hybrid TUI's design decision are summarised in Table 5.17. Each eQTL task would either be assigned a physical or digital control to increase users' focus on the tasks at hand and to aid motor-cognitive abilities. Well-established interactions for navigation and selection would be assigned to metaphorical and abstract gestures common in touch UIs. Verb objects would be contextually specified to represent gene expression files and subsets. Control of an object's facets would expand interactions to incorporate eQTL tasks. A gene expression objects would open a gene expression file, close or hide it from view. Gene expression objects would also be combined by stacking one object on top of the other. A SNP objects would filter gene expression files, collect filtered subsets and match significance across files. These interaction limit the number of objects required, adopts natural interactions to utilise motor-cognition, and ease manipulation.

5.5 Summary

In this chapter, design options for touch and tangible interfaces were considered for the development of an infovis TUIs for eQTL (Sections 5.2 and 5.3). The designs for the touch and tangible interfaces were combined to form a hybrid TUI, where each eQTL task was assigned to a digital or physical control. For tasks that required users to maintain their focus on the visualisation (navigation and selection), common metaphorical and abstract gestures were used. For the rest of the tasks, where motor-cognitive abilities are best utilised, objects were examined along a continuum of object meanings. Specific objects high in affordance were designated to containers (organise, compare, filter and match gene expressions), while verb objects tackled option selection (switch windows). The eQTL tasks and their hybrid interactions (see Table 5.17) were investigated in Chapter 6 using common TUI technologies tested for their technical feasibility.

Chapter 6

Infovis TUI Implementation

6.1 Introduction

This chapter first defines the modalities to be sensed by the TUI based on the case study's infovis tasks and their interactions (see Table 5.17). The following sections describe two common TUI technologies (micro-controllers with sensors and computer vision) and systematically evaluates their ability to sense the modalities. The technology best suited for this research's TUI is then used to implement an initial prototype. The TUI is informally evaluated to elicit subjective feedback about the system's usability and design. The feedback received is used to improve the TUI and touch UI baseline, which are described next. The chapter ends with a summary.

The system is envisioned to run on a multi-touch tabletop using touch for some interactions. Tangible objects are also utilised to represent gene expression files and significant subsets from the files' data sets. Along with the TAC number and the eQTL infovis tasks, Table 6.1 lists the modalities to be sensed by the system. Three common implementation technologies for indoor positioning, orientation and stacking include RFID, computer vision, and micro-controllers along with sensors and actuators (see Section 2.2.5). Touch interaction, of course, is supported by the multi-touch tabletop of our choice (a Samsung SUR40).

RFID is a radio-based wireless technology that uses electromagnetic signals to determine the presence and identity of tagged objects within the range of a tag reader. Micro-controllers are small computers that can be embedded in

TAC	eQTL Task	Sensing modality
1	Open/close files	Position
2	Switch windows	Orientation
3	Scroll, pan and zoom	Multi-touch
4	Select genetic variants	Multi-touch
5	Access external sources	Multi-touch
6	Organise windows	Position and orientation
7	Filter data	Orientation
8	Combine files	Stack
9	Match significance across files	Position

Table 6.1: TAC numbers (see Table 5.17), eQTL tasks and the modalities to be sensed by the TUI.

physical objects or environments to connect the physical and digital worlds. Using sensors and actuators, micro-controllers can receive information and affect the outside world. Computer vision is often used in TUIs because of its ability to sense multiple objects, particularly when fiducial markers are used. It can detect the position of an object, and sometimes its orientation, size and shape [210].

The following sections explore the technical feasibility of sensing the modalities using two of these technologies. RFID is not considered because of its inaccuracy at detecting position and orientation (see Section 2.2.5.1). Moreover, as objects are brought closer to each other (e.g. stacked or clustered), the performance of an RFID reader is likely to suffer considerably [146].

6.2 Micro-controllers' Sensors Technical Feasibility

This section focuses on the use of micro-controllers and sensors for developing tangible objects that could be used with planar surfaces. A reasonable number of electronic toolkits are available for prototyping (see Section 2.2.5.3). Of these

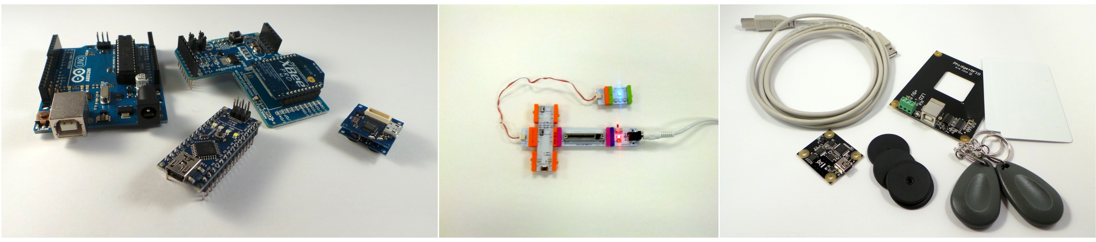


Figure 6.1: (left to right) Phidgets, littleBits, and Arduino.

toolkits, Phidgets, littleBits and Arduino (see Figure 6.1) were acquired to further assess their suitability.

Phidgets are a set of USB plug-and-play devices that do not require soldering of electronics [79, 80]. Developed at the University of Calgary, Phidgets have a wide range of applications [80]. The toolkit offers a large collection of different modules and sensors that are ready to use and are centrally controlled by a conventional computer rather than an external micro-processor. Some Phidgets are complete, self-contained sensing or actuating packages, while others serve as building blocks to be used with other sensors. Phidgets are programmable with various languages such as C, Java and Flash.

littleBits is a library of electronic modules that magnetically snap together to form prototypes [18]. Modules are known as bits and each serves a function, for example, light, buttons, pulse and threshold. There are four main types of bits: power, wire, input and output bits. The various bits do not require soldering, programming or wiring to create a more complex structure. A later addition, cloudBit, makes it easier to create interconnected devices and support the idea of the internet of things (IoT). While littleBits offers preprogrammed bits, some bits are programmable via their API.

Arduino is an open-source physical computing platform programmable through a language similar to C using the Arduino integrated development environment (IDE). It can operate either independently (i.e. standalone) or it can be connected to other Arduinos or a computer. Extension units, known as shields, can be plugged into an Arduino board to expand its capabilities. For example, a wireless shield can be used to communicate wirelessly with other modules.

Modality	Phidgets	littleBits	Arduino
Position	PhidgetSpatial 0/0/3 Basic)	Light sensor	Adafruit triple-Axis accelerometer
Orientation	Microload cell CZL616C	Light sensor	Adafruit L3GD20 triple-Axis gyroscope
Stack	Force sensor 1106_0	Pressure sensor	FlexiForce sensor

Table 6.2: Electrical prototyping toolkits and sensors that can be utilised to sense the modalities required of the infovis TUI.

Each of these toolkits provides a series of options for sensing the modalities in Table 6.1. Table 6.2, while not completely inclusive of all potential sensors, shows how each toolkit may be used with a certain sensor to detect an object's modality (position, orientation or stack).

Arduino proved to be the likeliest candidate for developing standalone wireless tangible objects whose position, orientation and stacking order can be detected using an accelerometer, a gyroscope and force sensors. Compared to the other micro-controllers examined, the Arduino offers adaptable micro-controller of varying sizes and extensive hardware and software support. Its online community is also the most active and the licenses for hardware reference designs are open source.

TinyDuino is a full-Arduino hardware platform with various expansion shields to add-on a multitude of sensors or lights (see Figure 6.2). A system is built with TinyDuino by snapping a number of miniature expansion boards (TinyShields) together. The sensors are easily programmable through the Arduino IDE. As a miniaturisation of Arduino, TinyDuino proved ideal for our purposes because of its size and stackable boards that minimised the need to solder sensors.

To develop a tangible object, the TinyDuino processor board is stacked with the following TinyShields: a USB and ICP board, proto board 1 and a WiFi shield. The following sections explore various means of detecting position, orientation and stacks. For each section a sensor is then chosen to technically evaluate.

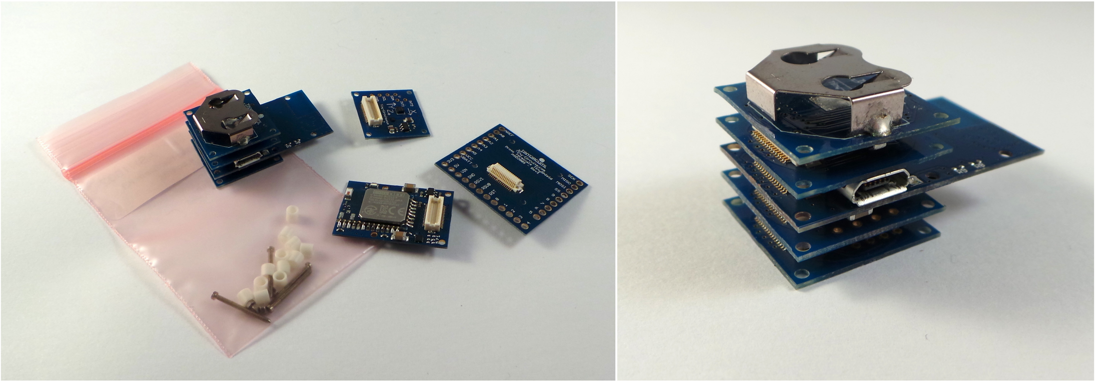


Figure 6.2: (left) A TinyDuino processor board and various TinyShields. (right) A stacked TinyDuino processor board with a USB and ICP board, wireless shield, prototyping board, and accelerometer.

6.2.1 Position

The spatial position of objects on a Samsung SUR40 can be detected along the display's x-and y-axes by using different sensory approaches. Ranging sensors emit a reference signal and compare the energy reflected with the one emitted. These types of sensors are commonly used for the detection of objects and their potential movement. Infrared (IR) sensors are capable of determining short distances by sending out an infrared beam and reading the reflection of the beam of the sensed object. IR sensors are capable of detecting ranges from 1.5 to 56 inches. Another range sensor for longer ranges, an ultrasonic sensor, sends an ultrasonic sounds and determines how long it takes for the signal to bounce back. Nevertheless, range sensors would require a clear field of view of the object in order no to obstruct the signal sent and received back [23].

An alternative but expensive option, uses magnetic motion trackers. These trackers give accurate readings for six degrees of freedom (i.e. could potentially detect stacks) using magnetic sensors which are attached to the objects meant to be detected [83]. Another options for measuring an object's position are accelerometers. Accelerometers are electromechanical devices that measure static and dynamic accelerations by sensing the object's movements and vibrations [21]. Of the various options considered, range sensors risk inaccuracy due to obstructions, whereby magnetic motion trackers are too expensive. Therefore, the following section evaluates the use of an accelerometer for the detection of an object's position on a 2D display space.

6.2.1.1 Accelerometer

Acceleration is the rate of change of the velocity of an object. Velocity is the rate of change of the position of that object. This means that the velocity of an object is the derivative of its position and acceleration is the derivative of the velocity. With each movement, there is an initial acceleration and deceleration until a maximum velocity is reached. This is then flipped the opposite way until it reaches rest. It is at this point that a new end position is reached. Therefore, the position of an object can be calculated by a double integration.

$$a(t) \tag{acceleration}$$

$$v(t) = v_0 + \int_0^t a\ dt^{'} \tag{velocity}$$

$$p(t) = p_0 + \int_0^t v\ dt^{'} \tag{position}$$

The MPU6050 device combines a three-axis gyroscope and a three-axis accelerometer with an on-board digital motion processor (DMP) that uses a standard I2C bus for data transmission. I2Cdevlib is a library developed by Jeff Rowberg for accessing MPU6050 and other I2C devices [198]. By utilising the hardware buffer on the chip and the DMP capabilities of MPU6050, the library performs data conversion between different coordinates and combines data from multiple sensors. This is valuable to obtain greater precision. The MPU6050 device along with TinyDuino satisfies the low power, low cost and high performance requirements of a wireless tracking system that may be used in this implementation to determine the position and orientation of tangible objects.

Prior to use, each MPU6050 device was calibrated to remove the acceleration offset component in the sensor output that was caused by the Earth's gravity. This was achieved using an Arduino sketch that averaged a collection of readings with five acceleration units until convergence (see Algorithm 1). The MPU6050

Algorithm 1 Calibrate the MPU6050 device

```
function Calibrate
    Initialise serial communication and device
    Establish and verify connection
    while not converged do
        $mean\_accel, mean\_gyro \leftarrow$CalcMean($buffer$)
        $offset\_accel, offset\_gyro \leftarrow$Calibrate($mean\_accel, mean\_gyro$)
    end while
    return $offset\_accel$, $offset\_gyro$
end function

function CalcMean($buffer$)
    Calculate $mean\_accel, mean\_gyro$
    return $mean\_accel, mean\_gyro$
end function

function Calibrate($mean\_accel, mean\_gyro$)
    $offset\_accel \leftarrow mean\_accel \div accel\_scale\_factor$
    $offset\_gyro \leftarrow mean\_gyro \div accel\_scale\_factor$
    return $offset\_accel, offset\_gyro$
end function
```

device was first tested with an Arduino UNO by connecting it to the Arduino using a breadboard and jumper cables. The connections between the Arduino UNO and the MPU6050 device were established as follows (see Figure 6.3):

- 5V ⇔ VCC
- GND ⇔ GND
- A5 ⇔ SCL
- A4 ⇔ SDA
- 2 ⇔ INT.

With this setup, the acceleration axes (x, y and z) were calibrated once, and the results were used to remove the offset prior to collecting real-world acceleration and computing position.

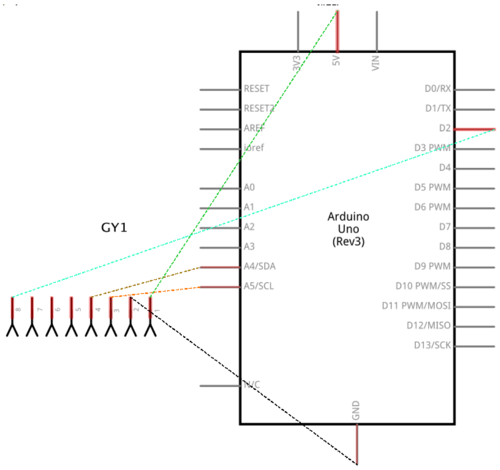


Figure 6.3: The MPU6050 device's connections to an Arduino UNO.

To construct the active tangible object, the MPU6050 device was soldered to a number of jumper wires that were in turn soldered to the TinyDuino stack's proto board 1 using the same connections described previously. The device was also connected to a lithium-ion polymer battery (see Figure 6.4).

Evaluation Method

The Samsung SUR40 display measured 88.77 cm × 50.02 cm in size. An area of the same size on an ordinary table was divided into a 9 × 5 grid, where each grid position was spaced at 10 cm intervals: 0-80 cm along the area's x-axis and 0-40 along its y-axis. The accuracy of the position calculations was determined by moving the object a total of 225 times (five trials for each of the 45 positions).

The object was fitted to a button collection circuit so that the button could be pressed to indicate the start and end of a trial. In each trial, the object was placed at the movement origin and the button was pressed. The object was then moved the required distance in an arc to its new position, and the button was pressed again to indicate the end of the trial. The acceleration component $aaWorld$ from the DMP6 library computed an object's acceleration with gravity removed and adjusted for the world frame of reference.

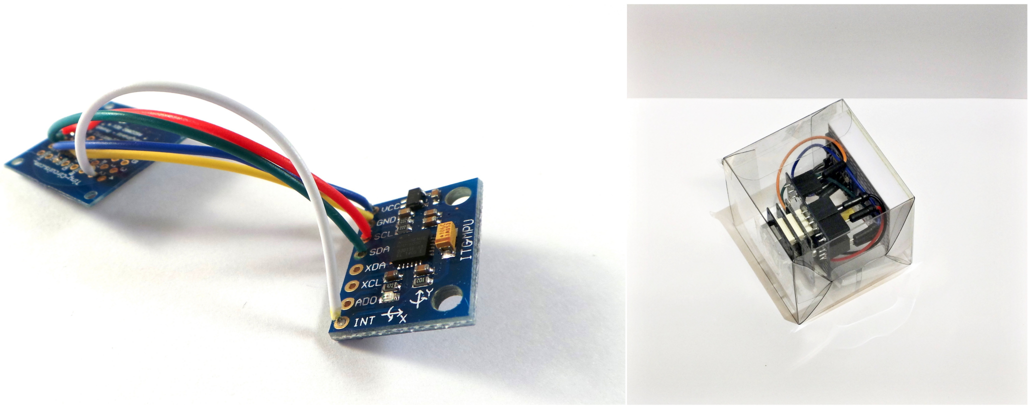


Figure 6.4: (left) The MPU6050 device soldered to a prototype board. (right) A tangible object consisting of a TinyDuino processor board, USB and ICP board, wireless shield, prototyping board and the MPU6050 device.

Once the corrected acceleration data were collected, noise was reduced in three ways [52] (see Algorithm 2):

First A frequency analysis of the data was performed and used to design a low-pass Butterworth filter. The acceleration data were processed using two passes of the filter (6 Hz threshold), the second pass needed to prevent a phase shift.

Second The filtered data were recalibrated owing to a shift in the offset after the object was moved. The data was passed through a moving window of half a second to recalculate the offset and then recalibrated.

Third After filtering and recalibration, there were still small non-zero accelerations when a tangible object was stationary. These readings were made zero by computing the maximum and minimum values collected for the stationary state. Values that fell within that threshold were considered stationary.

Figure 6.5 shows the three noise reduction methods applied independently to an object's acceleration that was displaced along its x-axis by 30 cm. Figure 6.5e shows the result of applying all three methods to the acceleration data. The final figure shows displacement after integrating the acceleration twice.

Algorithm 2 Acceleration noise reduction method

```
function FILTER(acceleration)
    $acceleration \leftarrow$ Buttworth($acceleration$, 6)
    return acceleration
end function

function RECALIBRATE(acceleration)
    $acceleration \leftarrow$ MovingAverage($acceleration, win\_width$)
    return acceleration
end function

function STATIONARYTHRESHOLD(acceleration)
    while object is stationary do
        $min\_stationary \leftarrow min(stationary)$
        $max\_stationary \leftarrow max(stationary)$
    end while
    for value in acceleration do
        if $min\_stationary < value < max\_stationary$ then
            $value \leftarrow 0$
        end if
    end for
    return acceleration
end function
```

Results

An object's positions was computed from the corrected acceleration data, and after the application of the noise-reduction methods (see Table 6.3). For displacement along a single axis, the results indicate that the recalibrated acceleration was able to detect the position of the object up to 40 cm along the screen's width or height with a margin of error of less than 1 cm. It proved less accurate with displacements greater than 40 cm. The application of the stationary threshold had little effect on accuracy.

Simultaneous movements along the x- and y-axes decreased the distances with negligible error from 40 cm to 20 cm (see Figure 6.6). For large movements, the error was sometimes greater than 10%.

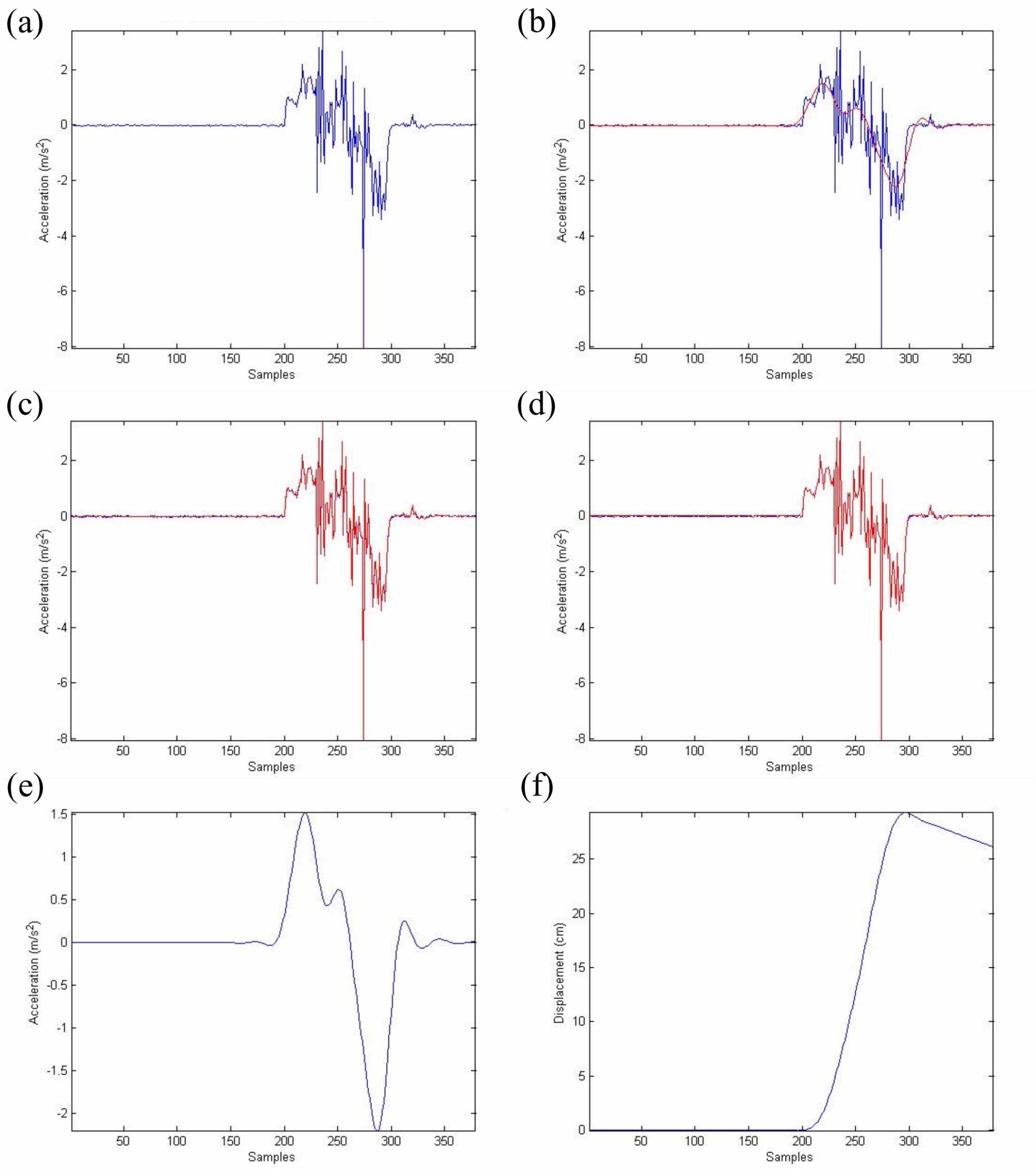


Figure 6.5: Computing 30 cm displacement along the x-axis. (a) Acceleration with gravity removed. (b) Acceleration (blue) and filtered acceleration (red). (c) Acceleration (blue) and recalibrated acceleration (red). (d) Acceleration (blue) and acceleration after applying a stationary threshold (red). (e) Acceleration after applying all three noise reduction techniques. (f) Displacement at 29.28 cm.

Axis	Target Distance (cm)	Mean (SD) position (cm)		
		Low-pass filter	Recalibrated	Stationary threshold
X	10	18.64 (7.36)	9.63 (1.15)	9.49 (1.28)
	20	16.04 (1.91)	20.51 (1.12)	20.47 (1.21)
	30	23.21 (3.24)	30.16 (2.13)	30.19 (2.12)
	40	37.29 (7.27)	39.91 (1.67)	39.73 (1.74)
	50	42.67 (7.03)	46.93 (5.11)	47.09 (5.11)
	60	51.15 (6.38)	62.08 (5.73)	61.78 (5.58)
	70	64.76 (4.23)	72.96 (3.94)	73.06 (3.93)
	80	63.36 (18.84)	83.61 (2.18)	83.33 (2.36)
Y	10	43.13 (13.58)	10.86 (2.76)	10.99 (2.72)
	20	58.23 (9.04)	20.33 (1.46)	20.19 (1.47)
	30	68.18 (10.88)	30.05 (2.69)	30.11 (2.57)
	40	82.48 (7.22)	40.89 (1.76)	40.49 (1.88)

Table 6.3: Mean and standard deviation (SD) of an active object's position after the application of each of the three noise-reduction methods to eight positions along the screen's width and four positions along its height.

6.2.2 Orientation

Various sensors can be used with a micro-controller to determine the orientation of an object with respect to other objects or space. Encoders sense rotations of an objects by combining a rotating wheel with slits and a light sensor. With an encoder, a micro-controller counts pulses emitted by the light sensor as they pass through the wheels slits to determine the wheel's rotation. Potentiometers are another option for sensing an object's rotation using voltage divider used for measuring electric potential [21].

While encoders and potentiometers are valid options for detecting an object's rotation, their bulkiness could potentially increase the size of our examined object. Alternatively, a more compact option uses an accelerometer along with a compass or a gyroscope to determine an object's orientation. This is possible since an accelerometer is capable of detecting static acceleration against gravity's

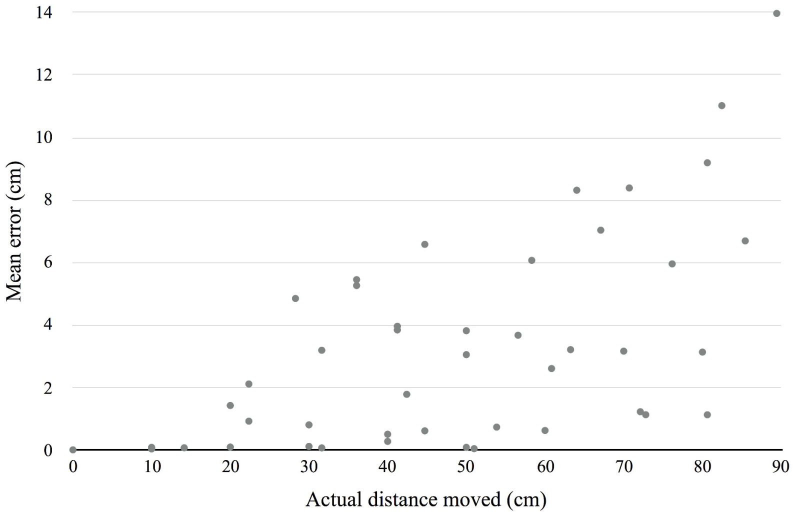


Figure 6.6: Actual dual-axes displacements and mean error as an object was moved to 45 positions along the surface's width and height.

reference. The following section technically evaluates the use of an accelerometer and gyroscope for the purpose of determining an object's orientation along its x-, y-, and z-axes.

6.2.2.1 Accelerometer and Gyroscope

To determine the orientation of a tangible object, the MPU6050 accelerometer and gyroscope data were combined. This was because the accelerometer results provide accurate orientation data if gravity was the only force acting on the sensor, but not when the device was being manipulated by a user. Raw data from the gyroscope include angular acceleration, but this is subject to drift over time. Therefore, to reduce noise from the accelerometer and drift from the gyroscope, the accelerometer and gyroscope data were fused using a proprietary algorithm that is part of the MPU6050 device's programmable DMP sketch.

Prior to use, the MPU6050's accelerometer and gyroscope were calibrated to remove the offset component in the sensor's output that was caused by the Earth's gravity. This was achieved using an Arduino sketch (see Algorithm 1) that averaged a collection of readings for until convergence.

Axis	Rotation (degrees)	Mean (SD) angular error (degrees)
X	90	8.84 (<0.1)
	180	<0.1 (<0.1)
	270	8.84 (<0.1)
	360	<0.1 (<0.1)
Y	90	8.84 (1.81)
	180	<0.1 (<0.1)
	270	11.81 (4.06)
	360	<0.1 (<0.1)
Z	90	13.29 (4.06)
	180	<0.1 (<0.1)
	270	17.25 (2.23)
	360	<0.1 (<0.1)

Table 6.4: Mean and standard deviation (SD) angular error when an active object was rotated around the x-, y and z-axes.

Evaluation Method

The calibration offsets calculated with Algorithm 1 were used in a quaternions sketch to calibrate the sensors. A series of 65 readings were recorded as the object was manipulated at 13 different orientations around the x-, y-, and z-axes in the following order:

- Initial reading when the object was static.
- 90°, 180°, 270° and 360° rotations around the x-axis.
- 90°, 180°, 270° and 360° rotations around the y-axis.
- 90°, 180°, 270° and 360° rotations around the z-axis.

For each position, five orientation readings were recorded in quaternions when the object was placed horizontally on top of a printed square. This ensured that the object was aligned with the environment's x-, y- and z-axes.

Results

For each reading, the error was calculated as the angle between the reading and the target orientation along the shortest path of rotation. The mean error ranged from zero (i.e. perfect accuracy) to 17.25° for z-axis rotations of 270° (see Table 6.4). These results indicate that the MPU6050 device always correctly detected the cardinal orientations of the cube's faces when the object was rotated.

6.2.3 Stack

There are a number of approaches that can be used for the detection of stacked objects. Adopting a serial peripheral interface (SPI) allows micro-controllers to communicate instead of utilising sensors. Nevertheless, sensors can still be shared among the communication micro-controllers. SPI is a communication interface that is commonly used in micro-controllers to enable them to communicate with peripheral devices. The protocol applies a master and slave approach to communication, where a single master initiates all communications with other slave devices. A SPI operates on single master protocol and alternatively a multi-master protocol can be used (e.g. I^2C). Although these protocols are well established, limitations include latency and master-slave transaction delays.

There are also various sensors that can be adapted for the detection of stack of objects. Magnetic reed switches have been used previously to successfully detect stacks with Stackables [133]. Magnetic reed switches are commonly used as proximity sensors but rely on proper alignment to accuracy detect stacks. Force sensitive resistors are an alternative that detects the changing weights of objects stacked up on it. It was reported with the implementation of Stackables [133] that the sensitivity of the force sensors caused inaccuracy when detecting stacks when users place their hands on the stack. The following section evaluates the use of force sensors to detect stacked objects.

6.2.3.1 Force Sensitive Resistors

FlexiForce force sensors were attached to each tangible object to determine if an object was placed on the sensor, i.e. stacked. The force sensor was first tested with an Arduino UNO by connecting it to the Arduino with a breadboard and

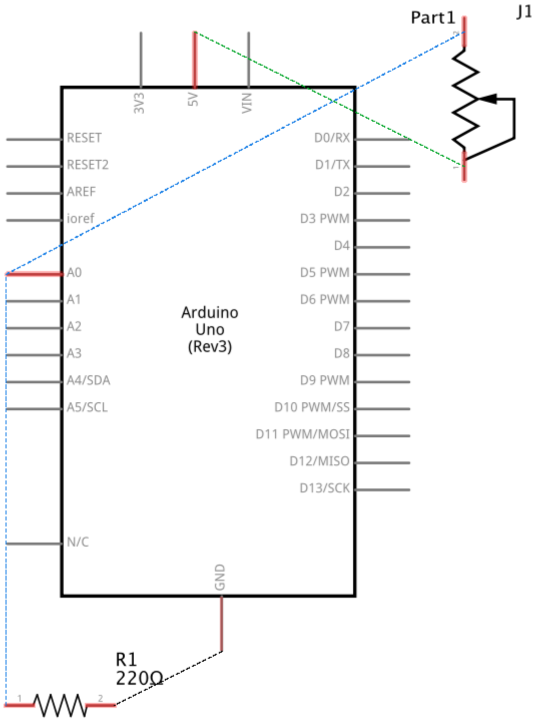


Figure 6.7: A FlexiForce sensor connected to an Arduino UNO.

jumper cables. The force sensor has a circular sensing area of 1 in diameter. To establish a connection between the Arduino and the force sensor, resistors are used to connect to a power source and an analog link (see Figure 6.7). When an object is placed on its sensing area, the resistance changes owing to the pressure applied. This indicates that the current flowing through the resistors increases which causes the voltage to increase. However, drift and hysteresis can affect the results. Drift is the change in the sensor output when a constant force is applied over a time period and is inversely proportional to time. Hysteresis is the difference in the sensor output when loading and unloading the same force. To minimise the effects of drift and hysteresis, the sensors were conditioned and calibrated.

To condition a force sensor, 110% of the maximum test load was placed on top of the sensor for a few seconds. The object in Figure 6.4 weighed 50 g, and therefore, a cube object of the same size and weighing 55 g was placed on the sensor for conditioning. Once the sensor was conditioned, three objects weighing

17 g, 33 g and 55 g were used for calibration. First, the 17 g object, which is a third of the object's weight, was placed on the sensor for 30 s and the sensed force recorded. The first weight was removed and two thirds of the object's weight, 33 g, was placed on the sensor for another 30 s and the output recorded. Finally, the entire object's weight was placed on the sensor for the same amount of time and recorded. Force versus resistance was then plotted and the best-fit curve was computed using the recorded data:

$$r = 0.1008 \ln(weight) - 0.2789,\ R^2 = 0.98 \qquad \text{(resistance)}$$

The equation for the line of best fit was then used to determine the weight of an unknown object during the evaluation.

Evaluation Method To evaluate the performance of the base object's sensor, four objects weighing 50 g were stacked on top of each other and readings were recorded. Naturally, the size of a combination in an eQTL study would depend on various factors, such as genes examined and the disease or trait of interest. For the purpose of this research, combinations are limited to five items (see Section 4.4).

First, the first object was placed on the base object and five resistance readings were recorded. Next, the second object was placed on the stack (base object and the first object) and five additional readings were recorded. The third and fourth objects were stacked similarly and each time five readings were recorded. Each set of five readings was averaged.

Results

For each reading, the weight of the objects was computed using the line of best fit equation. A reading was considered correct if the weight was within 20 g of the correct weight. Accuracy ranged from 0 to 100% (see Table 6.5). Contrary to expectations, the readings were only reliable for one object. However, given that each object would contain a force sensor, the force and position data could be combined to determine which files should be drawn in the same plot.

Tangible objects in stack	Mean (SD) weight (grams)	Percentage correct
1	52.44 (12.02)	100
2	109.81(14.06)	80
3	144.73 (17.08)	40
4	181.98 (53.17)	0

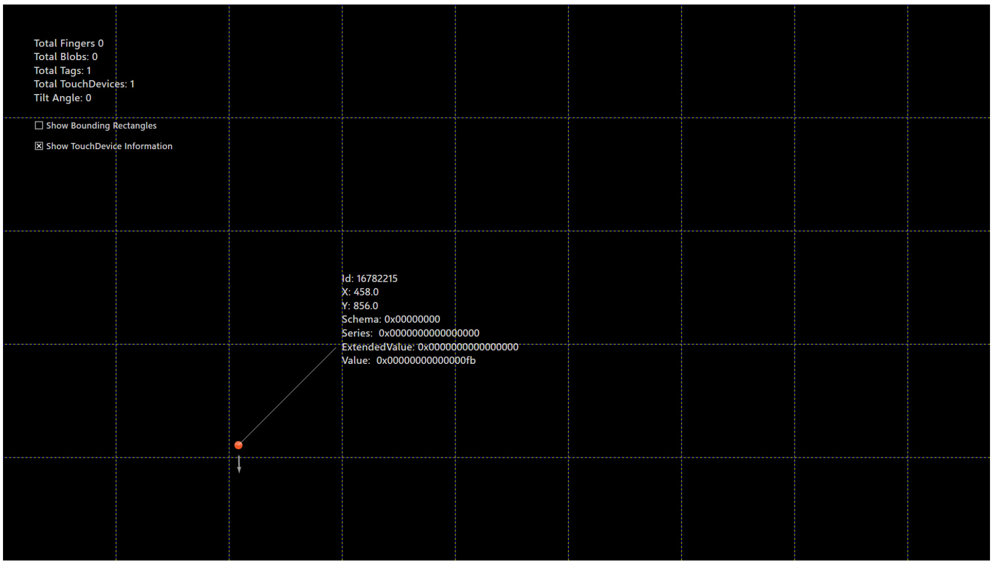
Table 6.5: Mean and standard deviation (SD) weight as one to four objects were stacked on a base object.

6.2.4 Discussion

An active tangible object utilising a micro-controller, accelerometer, gyroscope and force sensor was systematically evaluated to determine its accuracy given the three sensing modalities identified from the requirements: position, orientation and stacks (see Table 6.1). Using an accelerometer the position of an object was accurately detected for single axis displacements not exceeding 40 cm with a margin of error of less than 1 cm. For dual-axes displacement exceeding 20 cm, the error was sometimes greater than 10%. The disparity of the results from single axis and dual-axes displacement indicate that more noise is accumulated and amplified when moving the object diagonally from the starting position to the target. The face upon which a cube object was placed could always be correctly determined by combining readings from an accelerometer and a gyroscope. However, for dial-like rotations, higher accuracy is needed to correctly calculate the threshold value. Force sensitive resistors only proved reliable for the detection of a single stacked object.

6.3 Computer Vision Technical Feasibility

MBV systems use objects tagged with fiducial markers to uniquely detect their position in real time. This technology is often used in TUIs classified as interactive surfaces (see Section 2.2.2) to detect an object's position, orientation, and stacking order (e.g [15, 16, 34, 195]). The following sections evaluate the performance of a



```
Total Fingers 0
Total Blobs: 0
Total Tags: 1
Total TouchDevices: 1
Tilt Angle: 0

☐ Show Bounding Rectangles

☒ Show TouchDevice Information

Id: 16782215
X: 458.0
Y: 856.0
Schema: 0x00000000
Series:  0x0000000000000000
ExtendedValue: 0x0000000000000000
Value:  0x00000000000000fb
```

Figure 6.8: The Samsung SUR40's screen divided into a 9 × 5 grid using the Surface SDK 2.0 data visualiser sample application.

rear-projected MBV (the Samsung SUR40) at detecting the modalities required of the infovis TUI (see Table 6.1).

6.3.1 Position

The multi-touch tabletop of our choice (a Samsung SUR40) used MBV technology to detect the position of multiple objects on its surface as well as orientation. The surface supported two types of tags: coded byte tags and expandable identity tags. A byte tag was attached to a cube acrylic object to determine its spatial displacement across the x- and y-axes.

Evaluation Method

The Samsung SUR40 screen measured 88.77cm × 50.02 cm in size. The screen was divided into a 9 × 5 grid using Surface SDK 2.0 data visualiser sample application (see Figure 6.8). Each grid position was spaced at 10 cm intervals: 0-80 cm along the x-axis and 0-4 along the y-axis. The object was moved a total of 225 times, five displacements for each grid position.

Axis	Target position (cm)	Mean (SD) position (cm)
X	10	10.13 (0.04)
	20	20.33 (0.02)
	30	30.48 (0.07)
	40	40.65 (0.04)
	50	50.81 (0.1)
	60	60.94 (0.04)
	70	70.97 (0.05)
	80	81.12 (0.04)
Y	10	10.23 (0.05)
	20	20.33 (0.03)
	30	30.46 (0.05)
	40	40.59 (0.05)

Table 6.6: Mean and standard deviation (SD) of a tagged object's displacement to eight positions along the screen's width and four positions along its height.

For each trial, the object was placed at one of the 45 positions and the position value was recorded in WPF independent variable unit and then converted to centimetres (*position_cm*) using the following equation:

$$position\_cm = (position\_wpf \times 2.54) \div PPI \qquad \text{(position in cm)}$$

Where *position_cm* is position in cm and *position_wpf* is displacement in WPF independent variable unit. Pixel per inch (PPI) was dependent on screen size and resolution and was found to be 55 PPI for the Samsung SUR40.

Results

For each grid position the five trials were averaged (see Table 6.6). For movements along a single axis, the interactive surface was able to detect the position of the object rapidly with a margin of error less than 1 cm. This was also the case for dual-axes displacements.

6.3.2 Orientation

An object must be placed on the surface in order to be detected. Therefore, if an object's sides were tagged, then it may be rotated on its x- or y-axis and then placed on the surface to activate a new command or state (e.g. a cube object can have up to six distinct states). When in contact with the surface, a tagged object's orientation around its z-axis was also detectable.

Evaluation Method

For the detection of a tagged object's z-axis orientation, a series of 25 readings were recorded as the object was manipulated at five different orientations around its z-axis in the following order:

- Initial reading when the object was static.
- 90°, 180°, 270° and 360° rotations around the z-axis.

The Surface SDK 2.0 data visualiser sample application was adapted to include a rectangular shape matching the size of the tangible object to correctly align the object at each of the five orientations. For each orientation, five readings were recorded in degrees.

Results

The surface accurately detected the evaluated angles along the z-axis (see Table 6.7). For rotations around the z-axis, the interactive surface was able to detect the orientation of the object rapidly with a margin of error less than 1°.

Axis	Rotation (degrees)	Mean (SD) rotation (degrees)
Z	90	91.32 (1)
	180	180.54 (1.1)
	270	270.04 (0.6)
	360	359.92 (0.6)

Table 6.7: Mean and standard deviation (SD) rotation when a tagged object was oriented around its z-axes.

These results indicate that the MBV system always correctly detected the tagged object's orientation around its z-axis with negligible margins of error.

6.3.3 Stack

To be able to identify stacked objects with an MBV system, additional technologies were required (e.g. glass fibre bundles [16]). Fiducial markers could also be modified to detect stacks and their order [15]. Nevertheless, both approaches introduced restrictions to the object's shape (e.g. stacked objects are expected to be flat) and size. Another approach, StackTop [195], augmented a computer vision interactive surface with a projector and Kinect in order to detect stacked documents. Alternative techniques for stacking utilised capacitive displays instead of computer vision (e.g. [34]); however, they still imposed similar restrictions on stacking objects.

6.3.4 Discussion

MBV systems are generally robust and reliable with the ability to track a large number of tagged objects. An object's position was accurately detected rapidly with negligible margin of error (less than 1 cm), which can arguably be caused by the placement of the tag on the object. For the detection of the z-axis orientation, the tagged object also proved accurate. Although tag degradation was an issue, following printing guidance and protection (e.g. the use of UV varnish) could lengthen a tag's time of use. The modalities in Table 6.1 were largely supported by the Samsung SUR40, however the restrictions imposed on objects in order to detect stacking led us to consider another approach.

6.4 Initial Infovis TUI

The use of micro-controllers, sensors and actuators showed promise for sensing the position, orientation and stacking of an active tangible, however the current technologies and approaches undertaken in this research has proven insufficient to achieve some of the modalities envisioned for the TUI. The built-in capabilities of the Samsung SUR40 for multi-touch and marker detection largely support the TUI's interaction techniques in Table 6.1.

In order to detect stacks when forming combinations, too many changes may have been imposed on the objects. Alternatively, clustering can be utilised but falls at the risk of causing too much clutter when forming combinations. For these reasons, combinations are formed using a tapping gesture with the gene expression object (see Table 6.8).

The infovis TUI consists of three main components: an interactive surface, an eQTL visualisation application and a number of tangible objects. The Samsung SUR40 multi-touch tabletop with PixelSense technology is used as an interactive surface. The initial application is developed on the tabletop using C#, windows presentation foundation (WPF) and the Microsoft Surface SDK. The visualisations are implemented using OxyPlot, a plot generation open-source cross-platform library for .NET [182].

6.4.1 Tangible Objects

Gene expressions and SNPs are inherently abstract data, and thus, abstract objects are deemed suitable to act as containers and controls for these two types of abstractions. The shapes of the objects are determined by mapping a value or state to various types of surfaces [50, 188, 216]. For example, discrete options are ideally represented by flat surfaces, and continuous values are better represented with curved objects. The tangible objects are made reusable to minimise clutter. By economising object use, the interface can be scaled to reflect the large number of gene expression files that are investigated in eQTL studies.

Cube acrylic objects are used to represent gene expression files and act as both containers for the files' data sets and controls. Each face of the cube is mapped to a discrete value that is associated with a different mode: plot window, table window or combination control. To identify each of these values on the tabletop, double-sided fiducial markers are attached to each of those sides. Each cube side facing a fiducial marker is also identified using two iconic signs (table and plot) and one symbolic sign (combination) [59] representing its discrete values (see Figure 6.9).

TAC	Representation		Behaviour	
	Token	Constraints	Variable	Actions
1.1	❒	⊓	Open files	Place object on thumbnail
1.2	❒	⊓	Hide files	Rotate object and place it on its hide side
1.3	❒	⊓	Close files	Drag object to a graphical recycle bin
2	❒	⊓	Switch windows	Rotate object and place it on surface on one of its sides
3.1	✋	🗔	Scroll table	Press with fingers and move
3.2	✋	🗔	Pan plot	Press with fingers and move
3.3	✋	🗔	Zoom plot	Pinch and spread
4.1	✋	⊓	Select genetic variants	Tap on genetic variant
4.2	✋	⊓	Expose information	Double tap on genetic variant
4.3	✋	⊓	Deselect genetic variants	Tap on genetic variant
5.1	✋	⊓	Open external sources	Tap on external source
5.2	✋	⊓	Close external sources	Tap on close
6.1	❒	⊓	Translate windows	Move object on surface
6.2	❒	⊓	Rotate windows	Rotate object on surface
7	⌓	🗔	Filter data	Place object on window and rotate dial
8.1	❒	🗔	Add to combination	Tap or place object on another window
8.2	❒	🗔	Remove from combination	Tap or place object on another window
9	⌓	🗔	Match significance	Place object object on another window

Table 6.8: Hybrid TUI's core elements and interactions described using the TAC paradigm. The updated interaction is highlighted in yellow.

Cylindrical acrylic objects represent subsets of SNPs collected from a gene expression file (see Figure 6.10). SNP objects are used to filter gene expression

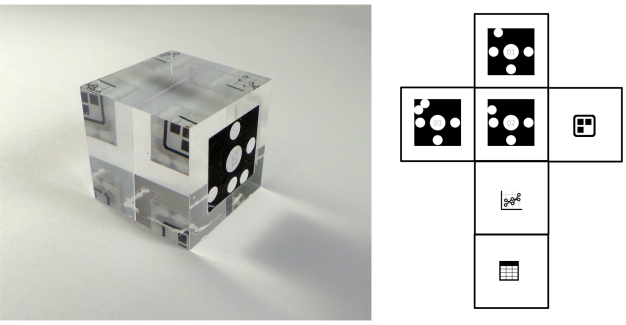


Figure 6.9: (left) A gene expression object made from a 3 cm clear acrylic cube. (right) The placement of the byte tags and signs on the object.

files and collect SNPs that can be viewed independently. The flat end of the cylinder is used as the viewing mode or a filtering/highlighting control based on its placement on the display. The convex surface of the cylinder is used to select a single value from the continuous threshold variable. The action of rotating the cylinder object imitates that of rotating a dial; an approach previously adopted in the infovis TUI literature (e.g. [9, 55, 169, 241, 245]).

The sizes of the cube and cylinder objects are decided by two factors: the size of the fiducial markers (1.91 cm × 1.91 cm square tag) used to identify the various values and states, and the users' ability to manipulate the object using one hand to encourage bimanual interaction and epistemic actions.

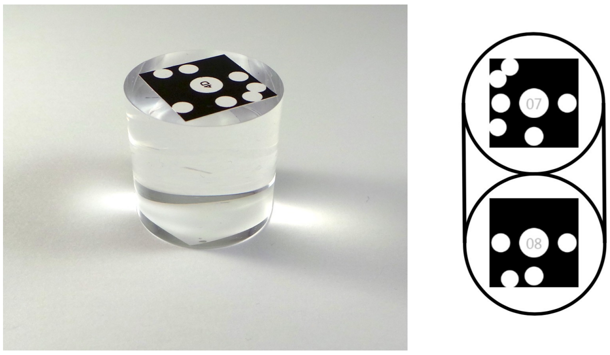


Figure 6.10: A SNP object made from a 3 cm diameter clear acrylic cylinder. (right) The placement of the byte tags on the object.

6.4.2 eQTL Visualisation Application

The interface is used as follows. A series of gene expression files are loaded and displayed as thumbnails in the file explorer at the top right corner of the screen (see Figure 6.11). The size of the thumbnail is determined from the size of the gene expression objects and is made large enough to aid the user discern general patterns prior to opening files (400 × 200 WPF independent variable unit). To expose more files in the file explorer, a user scrolls with one or more fingers.

In the top left corner lies the object recycler where gene expression and SNP objects can been placed to be cleared of their content and be reused (see Figure 6.11). Prior to any interaction, an action from the user is prompted with a message at the bottom of the screen on how to open a gene expression file. The message disappears as soon as the user interacts with the display. Information and warning messages appear at the bottom of the display to aid the user. After 30 s, the message dims and then disappears 15 s later.

Figures throughout the TUI's description are print screens collected with the visualisation application. The print screens show the digital response of the system as objects are manipulated. Gene expression and SNP objects are not pictured but are represented with square and circular shadows instead. The infovis TUI is shown in Figure 1.2.

6.4.3 Open/Close Files

To open a gene expression file, a user places a gene expression object on its thumbnail. The gene expression object has to be placed on its plot or table window side to link a gene expression file with the object. If an object is placed on its combination side, then a message appears at the bottom of the display that informs the user of the mistake and gives advice on how to proceed. When the file is successfully linked to an object, the system responds by dimming the thumbnail and a message appears declaring a successful link (see Figure 6.12).

The user temporarily closes a file by removing its object from the surface. To close a file permanently, the user places the file's object on the object recycler to empty its content. As the object recycles, the thumbnail associated with that gene expression file and object is reset to its original dimness. Because a gene

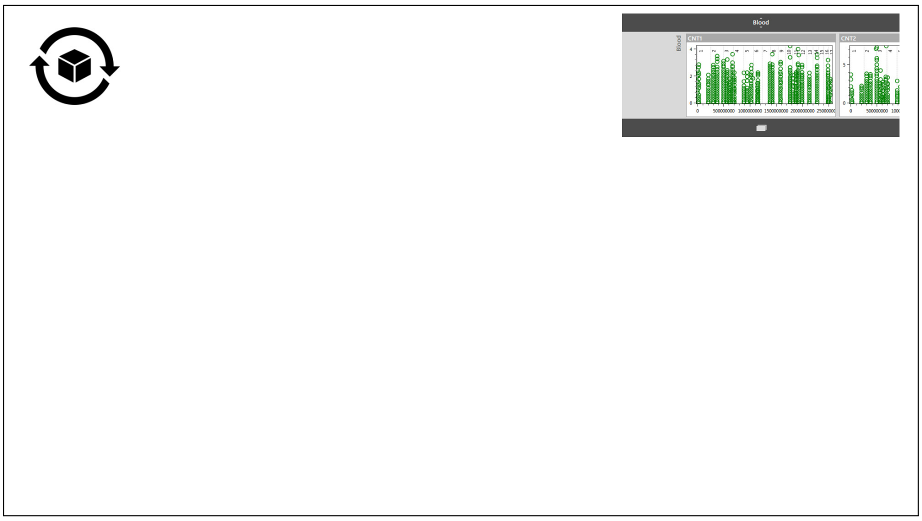


Figure 6.11: eQTL visualisation application with the file explorer and object recycler at the top right and left corners, respectively.

expression file can only be linked to one object, dimming a thumbnail disables any interaction with other objects.

6.4.4 Switch Windows

A gene expression window is displayed when its corresponding object is placed on the surface. The window consists of a circular identifier that highlights the object, a display area that shows the eQTL gene expression file and a reset button on the corner of the window. A user switches between the two file windows, a Manhattan plot and summary table, by orienting the gene expression object and placing it on one of its marked sides.

A Manhattan plot maps SNP data points to their genome-wide locations and their $-log_{10}$ significance. The plot is divided with a faint grey line into chromosomes and the data points are drawn as coloured rings on a white background. A red threshold line is also shown and initially set to zero. The table window displays a table of four columns: chromosomal location, SNP RSID, genome-wide base position and significance value. Figure 6.13 shows the display windows for the a gene expression file.

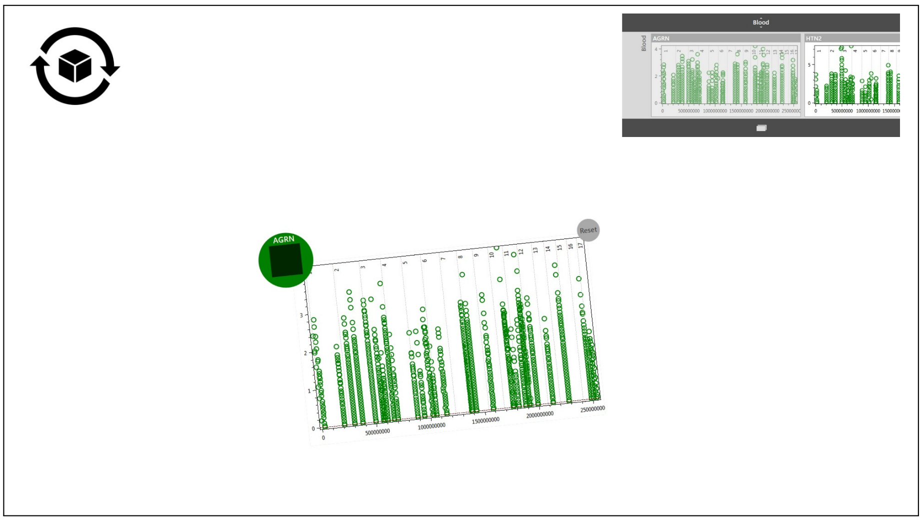


Figure 6.12: A gene expression object is placed on the surface (represented with a square) to display the file's window. The file's corresponding thumbnail is dimmed in the file explorer.

6.4.5 Scroll, Pan and Zoom

A gene expression file's Manhattan plot or table windows are zoomed, panned and scrolled to provide users with information that cannot be displayed at one time or to alter the users' viewpoints to increase their understanding of the results. Considering that a typical gene expression file contains tens of thousands of data points, navigation must be fluid and responsive.

To zoom out a plot, a user adopts a pinch gesture by touching the surface of the plot with two fingers and bringing them closer together. The inverse, touching the surface with two fingers (from one or two hands) and moving them apart, zooms in the plot. The user pans a plot by brushing the surface of the plot with one or more fingers with respect to the direction of the panning. To reset the zoomed and/or panned axes of the plot, the user taps with a fingertip on the reset button displayed on the corner of the plot window. In the table window, the user navigates the rows by scrolling. The user scrolls by brushing the surface of the table with one or more fingers with respect to the direction of the scrolling.

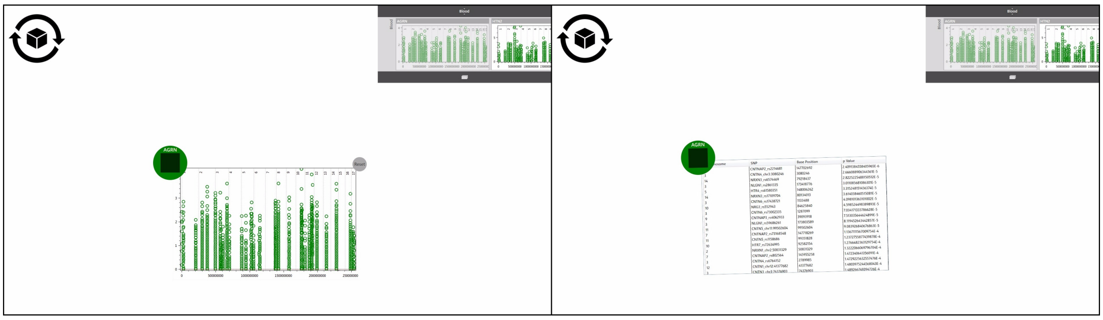


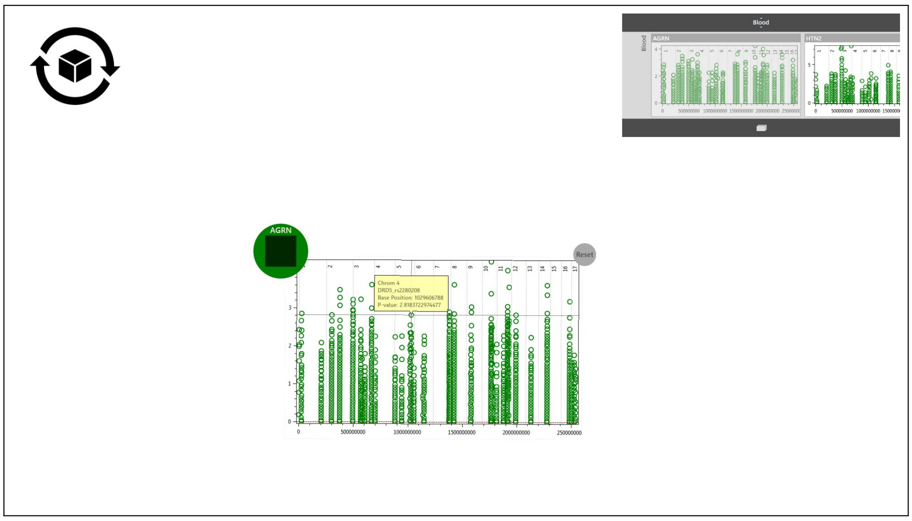
Figure 6.13: A gene expression file's plot (left) and table windows (right).

6.4.6 Select Genetic Variants

Genetic variants (SNPs) are selected in either plot or table windows to expose additional information. In a plot window, a user taps on a data point with a fingertip to select a SNP. To expose additional information about a SNP (infotip), the user taps twice. This information includes the SNP's RSID, chromosomal location, base position and $-log_{10}$ significance (see Figure 6.14). The infotip is closed when the user selects another SNP or after 60 s of no interaction with the information. Similarly, a user taps a table row with a fingertip to highlight it. A row highlight is removed when the user selects another row or after 60 s of no interaction with the table row.

6.4.7 Access External Sources

The user accesses external data sources (e.g. dbSNP or PharmGKB in Chapter 3) to gain knowledge about a certain SNP or provide explanation about its significance to a particular gene or disease. Additional information about a genetic variant is exposed by tapping a table row twice with a fingertip. This information includes access to external sources. The user taps on a source's link to open the external source in the table window. In this iteration, only access to dbSNP is provided and genetic variants are retrieved based on their SNP RSID. A user returns to the table window by tapping on the return button displayed on top of that window.



Chrom 4
DRD5_rs2280208
Base Position: 1029606788
P-value: 2.8183722974477

Figure 6.14: A genetic variant is tapped twice to expose additional information in an infotip.

6.4.8 Organise Windows

A user moves a gene expression or SNP object to move its digital representation within the viewing area. An object is either dragged across the surface (its digital representation follows the object's movement) or is picked up and placed at another position (this results in the digital representation being closed and then reopened once the object is replaced at its new position). Rotating the object around its z-axis orients the digital representation to make room for other windows or to share the object with collaborators around the tabletop.

6.4.9 Filter Data

The user works with a subset of a gene expression data set by filtering the data set using the SNP object. The user places the SNP object on its marked flat side on either a plot or table window. This actions superimposes the gene expression window with a circular digital dial. A digital arrow is also shown emerging from the bottom of the SNP object pointing upwards a significance threshold value of 0.0. The value of the significance threshold is displayed on the top-most side of the dial. To adjust the threshold, the user orients the SNP object clockwise or counter-clockwise around its z-axis. This in turn updates the displayed significance

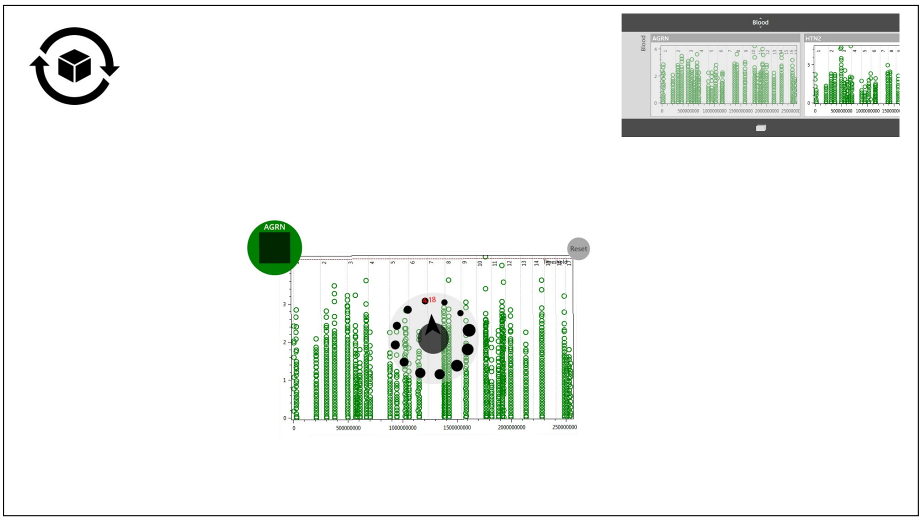


Figure 6.15: A SNP object is placed on a gene expression window to filter its data. The plot is superimposed with a digital dial displaying the significance threshold. The user rotates the SNP object to adjust the threshold.

threshold value. Figure 6.15 shows a SNP object placed on the plot window of a gene expression file, which exposes the dial (set at 4.18) and its components.

Rotating the SNP object collects a subset of SNPs from the gene expression file that meet the adjusted threshold value. The SNP object acts as a container for this subset which is displayed by placing the object anywhere within the viewing area. The new window consists of a gene expression and threshold identifiers that highlight the object and a display area that shows the subset of data presented in a table. A SNP object is emptied of its contents to be reused by placing it in the object recycler.

6.4.10 Combine Files

To explore patterns shared by gene expression files, a user combines up to five files to identify and examine their shared patterns. One of the gene expression object's sides is dedicated to combining its content with other gene expression files. In a gene expression combination, there exists a file that is used as the base of a combination to which other files are added or removed.

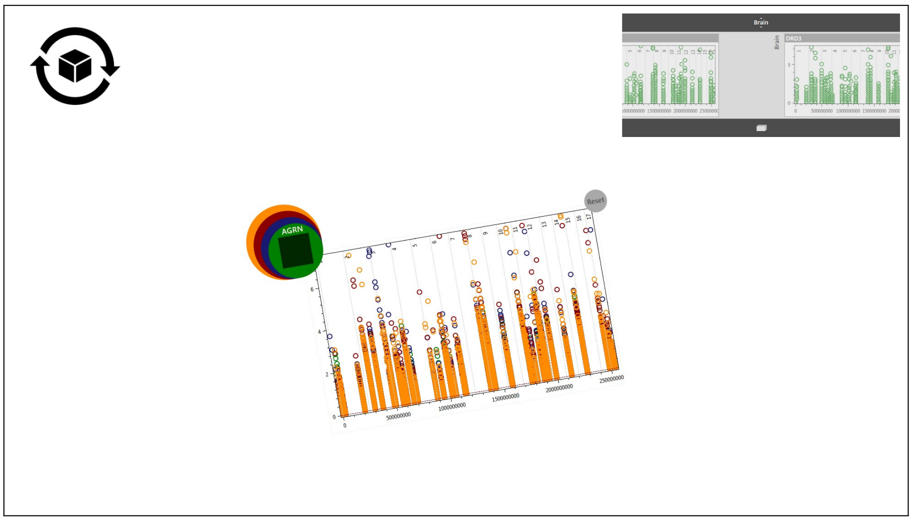


Figure 6.16: Four gene expression files are combined in a plot window. Each data set is superimposed over the other based on the order in which it has joined the combination.

To add a gene expression file to a combination, the user taps the combination side of its object on the plot or table window of a base file. This combines the two data sets into a larger data set, with both gene expression files uniquely identifiable by colour. It also updates the identifier circumventing the base object to include the name of the new gene expression file (see Figure 6.16). To remove a file from a combination, the user taps the gene expression object of that file on the base file's window. A gene expression file can be added to more than one combination, but cannot be added more than once for a certain combination. When a gene expression object is recycled, its content within combinations is also removed.

6.4.11 Match Significance across Files

Patterns within gene expression files are also detected by matching genetic variants collected from one file with another. A user can also match significance to test combinations prior to forming them. To match significance across gene expression files, a user first collects a subset of genetic variants from one file via filtering (see Section 6.4.9). The user then places the SNP object on another file's plot or table

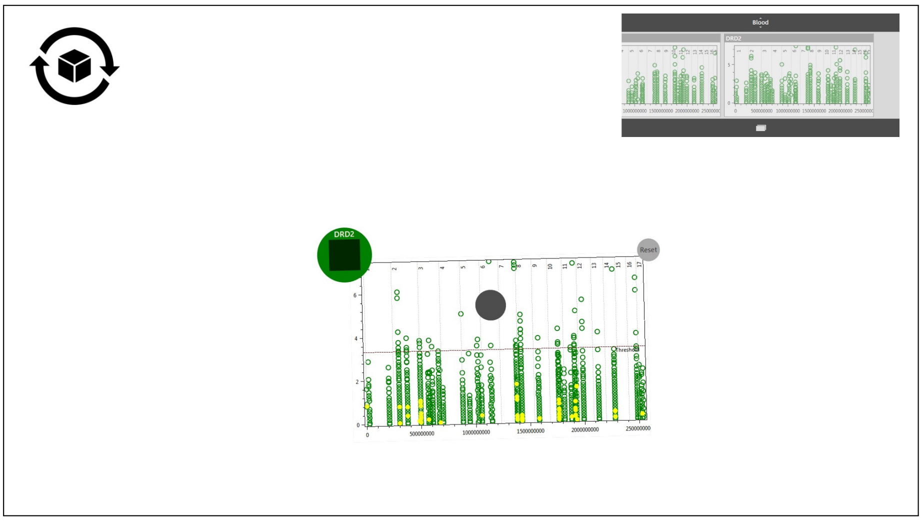


Figure 6.17: A SNP collection from one file is matched against the displayed gene expression data set. Matches are highlighted in yellow.

window. This action highlights the genetic variants contained within the SNP object in the file's data set (see Figure 6.17).

6.4.12 Informal Evaluation

The initial TUI application was presented to three postgraduate students (P1, P2, P3), who were encouraged to try out the system and provide feedback relating to its look and feel, interactions and performance. The system and its various functionalities were explained and demonstrated to the students. They were then presented with a list of combinations and were asked to try out the combinations and report on the patterns and genetic variants that were identified.

The students found that the objects were of a good size and could easily be held and interacted with using one hand. P1 and P2 found the signs identifying the display options for the gene expression object to be helpful and self-explanatory (see Figure 6.9). Nevertheless, P3 suggested that adding colour to the iconic markers could make them easier to spot when orienting the object. P2 noted that when rotating the object, it would be more practical to know what display they should expect when orienting the object in a certain direction prior to actually moving it. In effect, this meant adding directions to the signs to make interaction with the gene expression object faster. The SNP object had no identifying signs, and P1 and P3 preferred the object to have markings similar to those used with the gene expression object.

While interacting with the gene expression objects, P2 stated that rotating the objects and orienting their digital representations was unnecessary for a single user. This was because single users were unlikely to move around the tabletop as all areas of the surface could be accessed from a single sitting position. Using the same rationale, when filtering using a SNP object, P3 noted that the threshold value should be easily viewed from the sitting position, i.e. at the bottom of the rotating dial and not at its current top position. P1 and P2, both right-handed users, found that their arms would at times obscure the display. This was because the display area was shown to the right of the identifier highlighting the gene expression object (see Figure 6.12). P3 was left-handed and did not have this problem.

After filtering a gene expression file using a SNP object, P2 noted that as soon as the object was removed, the plot or table was reset. P2 would have preferred to have the filtering display maintained for a certain time period before it was completely removed. P1 suggested highlighting the genetic variants that meet the threshold significance when filtering a gene expression file. This would also be the case for filtering file combinations, but instead highlighting variants that meet the threshold in all file combinations. While matching a subset collected from one gene expression file with another by placing its SNP object on its display, P1 thought that the SNP object should also be identified along with the highlighted genetic variants instead of having to memorise the details of the SNP object.

In the initial system, a gene expression file could either be displayed as a table or a plot. P2 suggested that including a minimised window that shows the file's name would make the object easily identifiable. This, of course, was considered but was not implemented for the initial system (see Table 6.8). All three students struggled slightly when opening a file by linking it to its object. The user needed to place an object on either of its display sides and not on its control side to link it to a gene expression file. This exception to the rule seemed to cause confusion. This was also the case when using the combination control to combine files. For instance, P1 and P2 tried to perform the action of combining files with the display side of the gene expression object.

When moving objects on the surface, digital flickering happened infrequently but was noted with the virtual representations. This could either be caused by the printed fiducial markers (e.g. due to the print quality, folded edges or its placement on the object) or the free area surrounding the marker. Because of the transparency of the acrylic, it was suspected that contact on top of the object might be detected by the surface. This was particularly noticeable with the SNP object.

6.5 Final Infovis TUI

The suggestions from the three students were carefully considered and several changes were made accordingly to the objects and the application's usability, features and performance. The following sections only reflect on these changes.

6.5.1 Tangible Objects

The shape and size of the objects described in Section 6.4.1 are maintained. However, the labelling used to guide users when interacting with the objects are updated to reflect the suggestions made by the users of the initial system. Colour is added to the gene expression object's iconic labels to make them easier to spot. The cube object can be rotated around any of its three axes to reach a certain file encoding. To ease the process of rotating the object, iconic directions are also added to the labels (see Figure 6.18). The SNP object is also improved by wrapping a label around its dial control edge.

The sides of the object to be detected by the surface are covered with black vinyl to form a consistent dark non-IR reflective background for the fiducial markers. The fiducial markers are printed on white vinyl paper and covered with a top coat of UV print varnish. The markers are placed at the centre of each side of the object to guarantee full contact with the surface.

For this version of the system, only single users are considered; therefore, gene expression and SNP objects' orientation capabilities are disabled. This means

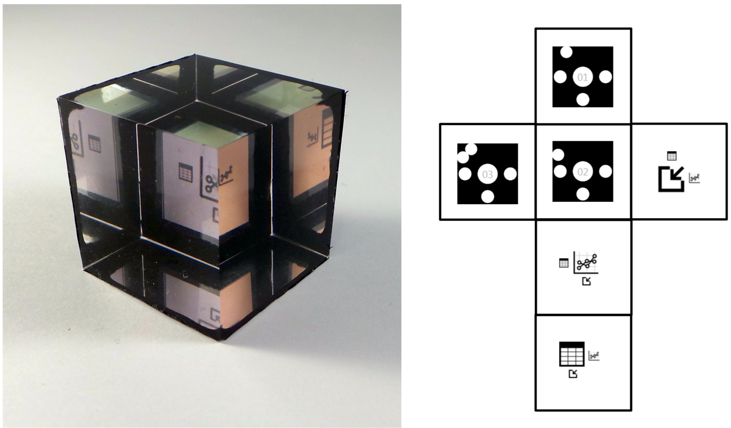


Figure 6.18: (left) A gene expression object with the new signs. (right) The placement of the byte tags and signs on the object.

that the object can be rotated around its z-axis while on the surface but its digital representation will remain in the same position.

6.5.2 Open/Close Files

Gene expression files in the initial system only has table and plot windows. An object's origin is only identifiable by fully displaying its content or by marking the object itself. A minimised window is included in this iteration, where the window's display area is collapsed with only the identifier rings displayed (see Figure 6.19). To open a gene expression file in the initial system, a user places a gene expression object only on its plot or table window sides to link the object to the file. After introducing the minimised view option in the current system, it is possible to link a gene expression file to an object by placing the object on any of its detectable sides.

The gene expression window consists of three distinct areas: an identifier ring that circumvents the physical object and displays the file's name, a display area that shows the file's content and a control button on top of the display area to reset navigation. The identifier is changed from a circle to a ring circumventing the object, with the name of the file printed within that ring's border. The original placement of the identifier is on the left of the display area, which obscured the

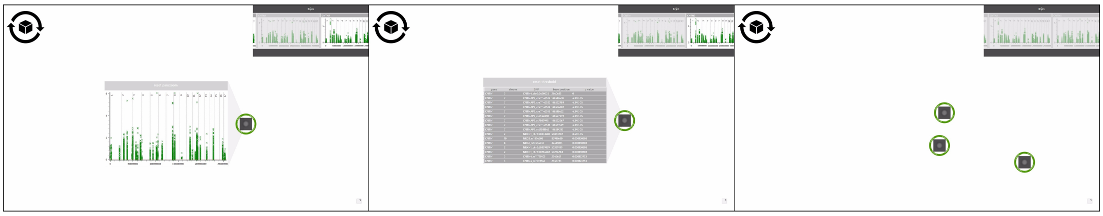


Figure 6.19: A gene expression file viewed asa plot (left) or a table (middle). A file is collapsed from view (right).

display when used by right-handed users. Therefore, the identifier ring is relocated to the right of the display area (see Figure 6.19).

6.5.3 Filter Data

When filtering gene expression files with a SNP object in the initial TUI, no feedback is provided to the user other than moving the threshold line across a plot. To view SNPs that meet the threshold requirement, the user places the SNP object on the surface to display its content. This has been adjusted to provide better feedback to the user when exploring gene expression files separately or in combinations. When filtering a gene expression file in a plot window, SNPs that meet the significance threshold requirement are highlighted in yellow. The threshold line is also adjusted to reflect filtering. When a gene expression file is displayed as a table, rows that do not meet the requirements are collapsed from view.

When filtering a combination of files, SNPs that meet the significance threshold in all files are highlighted in yellow in each of the files (i.e. if a combination of three gene expression files has one significant SNP, then three data points, one from each file, are highlighted). Similarly, in a table window, rows that meet the significance threshold in all files are kept in view while others are collapsed (i.e. if a combination of three gene expression files has one significant SNP, then three rows, one from each file, are kept in view; see Figure 6.20).

The SNP collection window is also improved to reflect the origins of the subset and the threshold value used for filtering its content. The new SNP collection window consists of gene expression and threshold identifier rings that circumvent

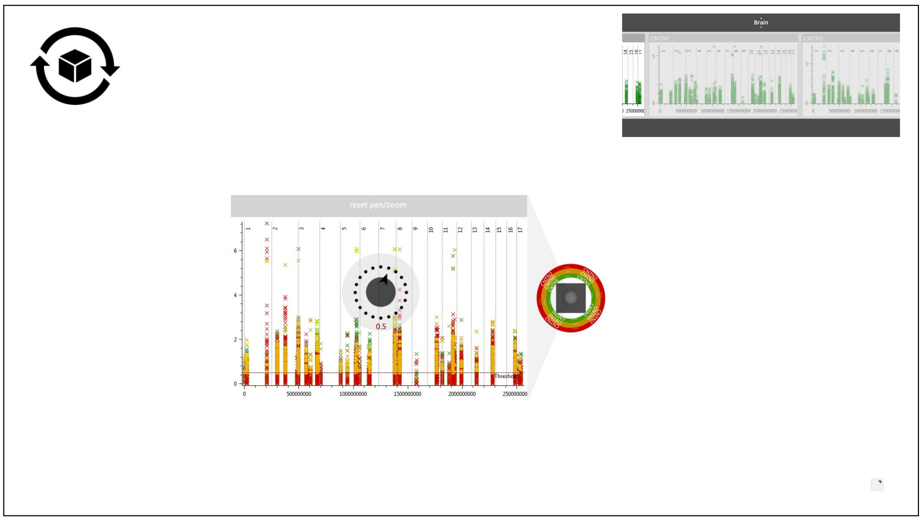


Figure 6.20: A combination of three gene expression files is filtered. Genetic variants that meet the threshold in all files are highlighted in yellow.

the physical SNP object. The display area also shows the subset in a simple list view (see Figure 6.21).

In the initial system, placing a SNP object on the surface of a gene expression window superimposes the window with a circular digital dial. The digital dial also displays the significance threshold on the top-most side of the dial. This is found to be inconvenient as the value is difficult to view from a sitting position. The position is relocated to the bottom of the rotating dial (see Figure 6.21).

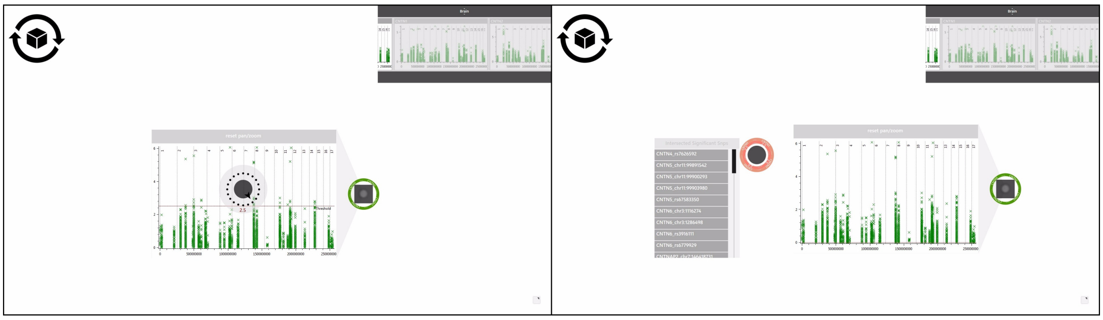


Figure 6.21: (left) A SNP object is placed on a gene expression window to filter the data set. (right) The SNP object is placed on the viewing area to display the list of genetic variants that meet the threshold requirement.

6.5.4 Combine Files

Shared patterns between gene expression files are identified by forming combinations via object taps. In the initial iteration, one side of the gene expression object is dedicated to adding or removing a file from a combination. The user taps once to add to a combination and taps again to remove from a combination. With the cube gene expression object, only three sides can be distinctly identifiable and detectable by the surface. With three viewing options (plot, table and minimised), the combination control will have to be shared with one of the window views in the current system. To avoid confusion, all detectable sides of the gene expression object are coupled with a combination control. To add a gene expression to a combination, the user taps with the gene expression object (with any of its detectable sides) on the window.

6.5.5 Match Significance across Files

With the initial system, a collection of SNPs are matched against another file by placing its corresponding SNP object on the other file's window. This highlights

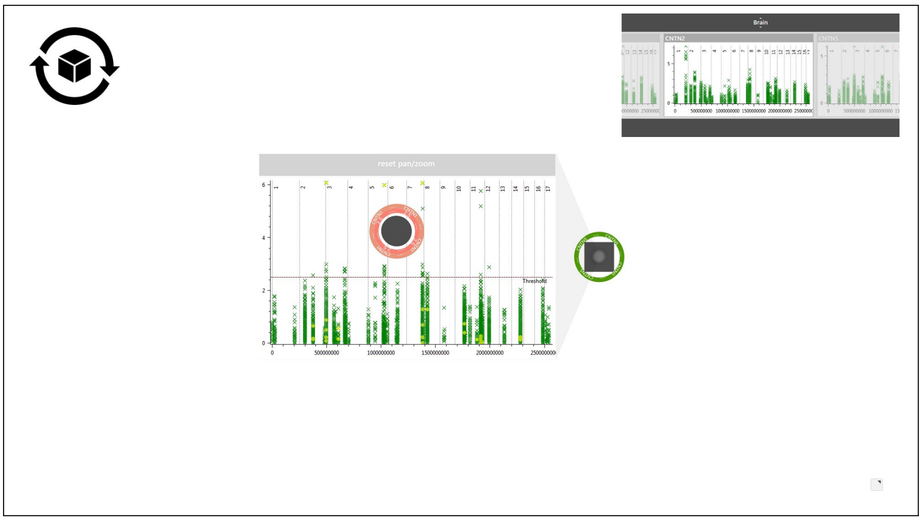


Figure 6.22: A SNP object houses a collection of SNPs from one file and is placed on the window of another file to highlight matches. Ring identifiers surround the SNP object to provide details about the subset being compared.

the genetic variants contained within the SNP object in the other file's data set. Also, the only feedback received from matching significance is the highlighted matches. This means the users have to keep track of the original content of the SNP object. To avoid this complication, the match display is updated to include two ring identifiers that identify the parent gene expression file and the filtering threshold (see Figure 6.22).

6.6 Infovis Touch UI

The touch UI combines desktop idioms with familiar metaphorical and abstract gestures (see Section 5.2). The WIMP menu metaphor combines functionalities that do not necessarily lend themselves to simple metaphorical gestures. In this iteration this includes: switching windows, combining files and matching significance. Common gestures are utilised for plot and table navigation. Continuous metaphorical gestures are adopted to opening and closing files (drag-and-drop), as well as filtering data sets. The multi-touch system is implemented on the Samsung SUR40 multi-touch tabletop with PixelSense technology. The application is developed using C#, WPF, and the Microsoft Surface SDK. The visualisations are implemented using OxyPlot [182].

The touch UI interface is used as follows. Gene expression files are loaded and displayed as thumbnails in the file explorer at the top right corner of the screen. In the file explorer, more files can be exposed by scrolling with one or more fingers. In the opposite top left corner of the screen lies the recycle bin, where gene expression files or SNP collections can be discarded (see Figure 6.23). As soon as the application is loaded, a message at the bottom of the screen prompts the user to open a file by providing instructions. Information and warning messages appear at the bottom of the display to aid the user. After 30 s, the message dims and then disappears 15 s later unless another message takes its place.

6.6.1 Open/Close Files

To open a gene expression file, the user presses on a file's thumbnail in the file explorer and drags the thumbnail into the viewing area. When the thumbnail is

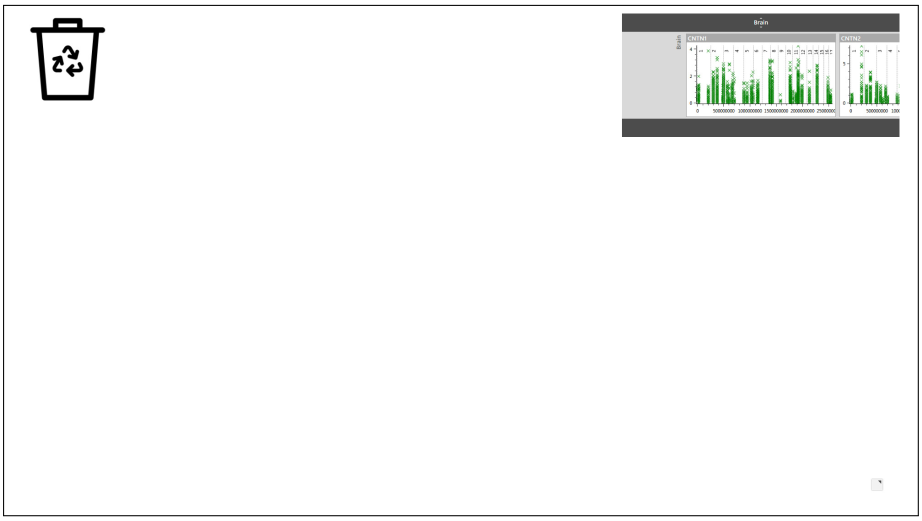


Figure 6.23: eQTL visualisation application with the file explorer and object recycler at the top right and left corners, respectively.

dropped in the viewing area, a gene expression file window is opened at the drop position. When the file is successfully opened, its thumbnail is dimmed and a message appears declaring a successful link (see Figure 6.24).

The user hides a file's window by selecting the option Hide from the View menu. This minimises the window to its identifier ring. A file window is maximised by tapping on the context menu anchored to the centre of the identifier ring and then selecting a window view to display by either tapping or pressing the option. The user closes a file by pressing the identifier ring and dragging it to the recycle bin. A discarded file's thumbnail is restored to its original dimness while its window is closed.

6.6.2 Switch Windows

When a gene expression file is dropped in the viewing area, a window of its content is displayed. Similar to the tangible interface, the window consists of an identifier ring stating the name of the gene expression file, a display area that shows the eQTL gene expression file data and a menu bar with view, combine or adjust threshold menu options. To switch between table and plot windows, the user taps

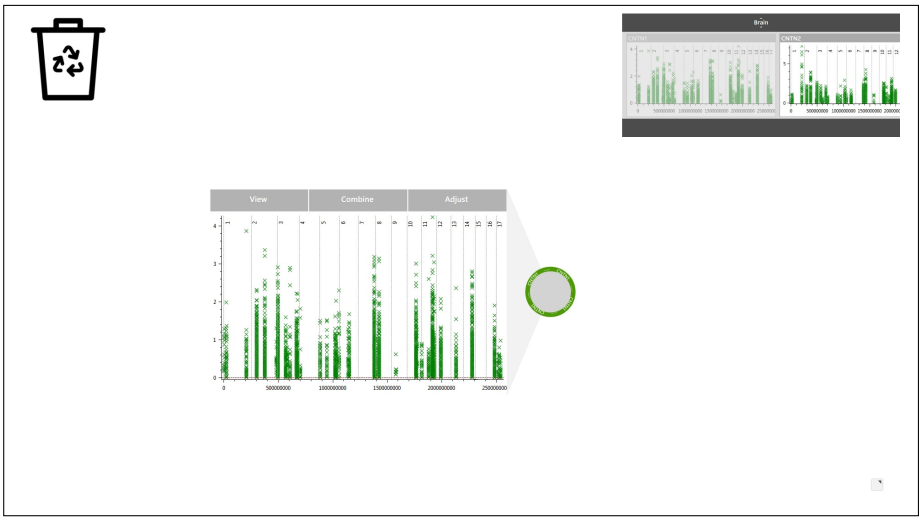


Figure 6.24: A gene expression file's window is displayed in the viewing area. The file's corresponding thumbnail is dimmed in the file explorer.

on the View option in the menu bar, and then selects either Plot or Table options to reload the data in that view (see Figure 6.25)

The plot window maps the SNP data points to their genome-wide locations and $-log_{10}$ significance in a Manhattan plot. The chromosomal locations are divided with a faint grey line, while data points are drawn as coloured crosses on a white background. A vertical threshold line is also drawn in red and initially set to zero. The table view displays a table consisting of five columns: gene name, chromosomal location, SNP RSID, genome-wide base position and significance value.

6.6.3 Scroll, Pan and Zoom

The plot and table windows of a gene expression file are zoomed, panned and scrolled to interact with more or less information. The techniques adopted to scroll, pan and zoom match those used for the hybrid TUI (see Section 6.4.5). In a plot window, the user zooms in and out using spread and pinch gestures with two fingers from one hand or a finger from each hand. Two-handed zooming can be symmetrical or asymmetrical. Plots are panned by brushing the surface of the plot with one or more fingers with respect to the direction of the panning. Zoomed

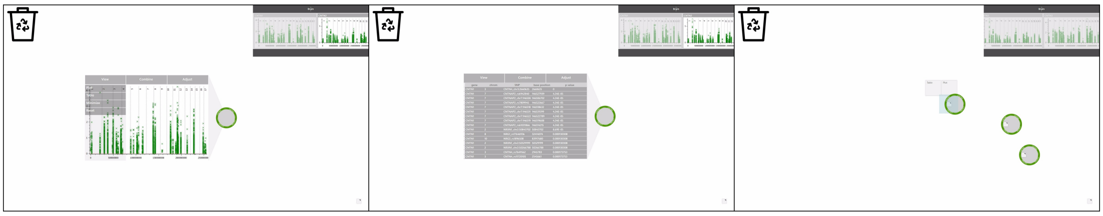


Figure 6.25: (left) Touch-based menu options to switch between views. (middle) A gene expression data set viewed in a table. (right) A gene expression window minimised to its ring identifier.

and panned plots are reset by tapping on the Reset option under View. In a table window, the user scrolls by brushing the surface with one or more fingers with respect to the direction of the scrolling. Tables can also be scrolled by navigating the scrollbar's thumb.

6.6.4 Select Genetic Variants

A user selects a genetic variant by tapping on the data point or row in a plot or table window. To expose additional information about a SNP, the user taps twice on the data point or row. Additional information is displayed in an infotip and includes SNP's RSID, chromosomal location, base position and $-log_{10}$ significance. The infotip either disappears after 60 s have passed with no interaction or when the user taps on another SNP.

6.6.5 Access External Sources

External data sources are accessed by the user to gain knowledge about a genetic variant or to provide explanation to its significance. To open an external source, the user exposes additional information about a genetic variant by double tapping on a row in a table window. The user then taps on a source's link to open the external source. The user returns to the table window by tapping on the return button.

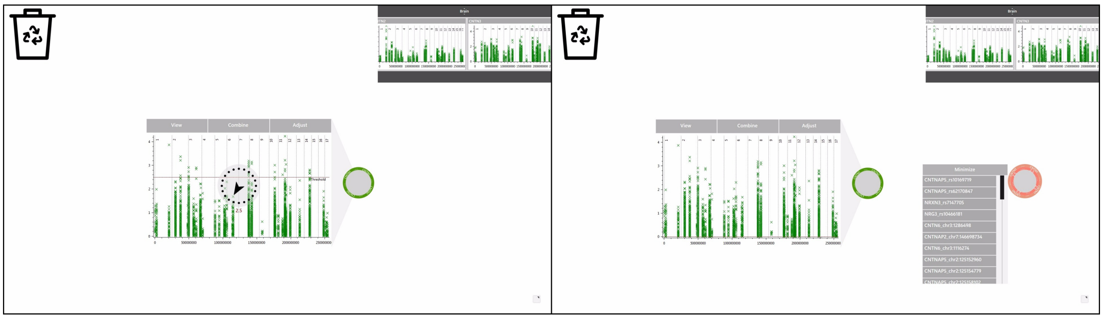
6.6.6 Organise Windows

Gene expression file windows are moved by pressing on or inside the identifier ring and dragging the window to the recycle bin. Rotating a window is possible with one or two hands, symmetrically and asymmetrically. However, this functionality is disabled since only single users are considered with this version of the system.

6.6.7 Filter Data

A user filters a gene expression file's data (either in a plot or a table window) by using the Adjust menu. The user taps on the options to superimpose the window with a circular digital dial and arrow. The arrow is navigated via a press and drag gesture to adjust the threshold. The adjusted threshold value is continuously displayed at the bottom of the dial display (see Figure 6.26). As the arrow is navigated, SNPs that meet the adjusted threshold are stored internally. In the plot window, the threshold line is also adjusted to reflect the change. In the table view, the rows that do not meet the threshold are collapsed from view.

The collected SNPs are viewed in an independent window when the user taps on the display option under Adjust. The window consists of identifier rings with the parent file's name in one ring and the threshold value in the second ring, a display area that shows the subset data set in a list view and a control button on top of the display area. As is the case with plot and table windows, subset windows are minimised by tapping on the minimise button. A minimised collection window



CNTNAP5_rs10169719
CNTNAP5_rs62170847
NRXN3_rs7147705
NRG3_rs10466181
CNTN6_chr3:1286498
CNTNAP2_chr7:146698734
CNTN6_chr3:1116274
CNTNAP5_chr2:125152960
CNTNAP5_chr2:125154779
CNTNAP5_chr2:125158102

Figure 6.26: (left) A gene expression file's significance threshold is adjusted. (right) The collected genetic variants that meet the threshold requirement are displayed in a list.

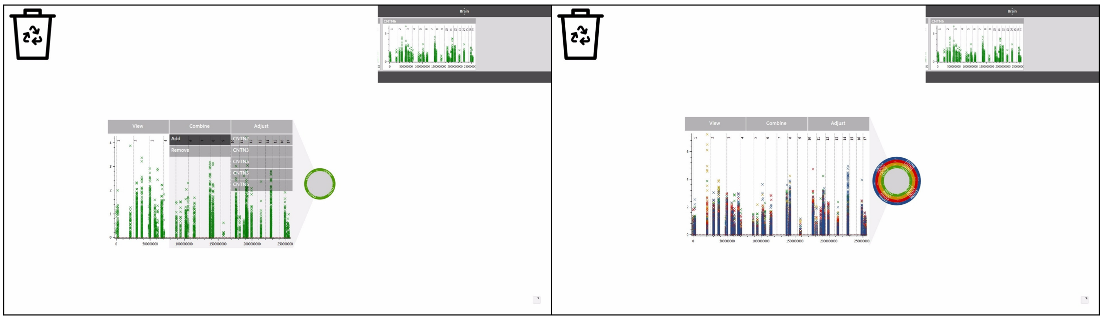


Figure 6.27: (left) Touch-based menu options to form combinations. (right) A base gene expression file is combined with three other files.

displays its identifier rings as well as a context menu anchored to the centre of the rings. To maximise the collection window, the user taps on the context menu and then the option to expose the list view. Collection windows can also be closed via drag-and-drop to the recycle bin.

6.6.8 Combine Files

A user can combine up to five gene expression files to identify and explore shared patterns. From the menu bar, the user taps on the Combine menu with the options to remove or add a file to the open window (see Figure 6.27). The Add option lists all the loaded gene expression files, while the Remove option only displays files within a combination. If a gene expression is part of a combination, then the corresponding option under Add is dimmed to deactivate interaction. Users add and remove files by tapping or placing a gene expression object on a window. A gene expression file can be added to more than one combination but can only be added once to the same combination. When a gene expression file window is dropped in the recycle bin, its content within a combination is also removed. The user is informed of any errors or mistakes detected by the system.

6.6.9 Match Significance across Files

Collections of genetic variants are matched against another file's data set to detect patterns between files. After filtering and collecting SNPs from one file (see Section 6.6.7), the user compares the subset against another file using the Compare menu

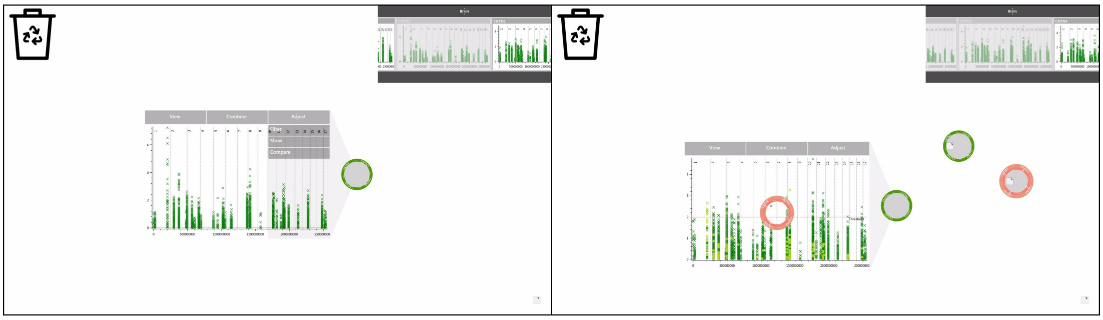


Figure 6.28: (left) Touch-based menu option to match significance across files. (right) A SNP collection from one file is compared against another gene expression's data set.

option under the Adjust menu. The Compare option lists the collections that can be compared with the file. A collection is selected via tap and its genetic variants are highlighted in the file's data set (see Figure 6.28).

6.7 Summary

In this chapter, sensing modalities were evaluated with two common TUI technologies to determine the most suitable implementation technology. Three sensing modalities were identified from the elicited functional requirements and the explored deign options for a hybrid TUI: position, orientation and stack. The performance of a micro-controller for sensing the modalities proved inadequate (see Section 6.2) compared to the performance of an MBV system (see Section 6.3). This approach was used to implement an initial TUI. The TUI system consisted of a Samsung SUR40, an eQTL visualisation application and tagged tangible objects (see Section 6.4). An informal evaluation was carried out with postgraduate students that were asked to explore the system. Feedback was gathered and used to refine the system into a final TUI (see Section 6.5). A touch UI was also developed (see Section 6.6) and used as baseline for evaluating the final TUI in Chapter 7.

Chapter 7

Infovis TUI Evaluation

7.1 Introduction

The overall aim of the research is to design, develop and evaluate a TUI for infovis, using eQTL analysis as a case study. The research hypothesises that a TUI system would outperform a touch UI supported by the theoretical foundations recalled in Section 2.2.3. This chapter describes a user experiment that compares the TUI and the touch UI. The goals of this experiment is as follows:

1. Examine the performance of the TUI and touch UI systems for identifying significant genetic variants in gene expression files and detecting patterns in combinations of these files.
2. Inspect users' interaction for bimanualism and epistemic actions.
3. Learn about participant preferences while using the TUI and touch UI.

In the experiment, participants are asked to explore gene expression files individually and in combinations to record significant variants. Participant's interactions and explorations were logged electronically and observed. At the end of the experiment, questionnaire answers and comments about the interfaces are collected. The results of the experiments are presented and discussed next. The chapter ends with a summary.

7.2 Method

The experiment adopted a between-participants design. We made the following hypotheses based on theoretical concepts that strengthen TUIs and past literature (see Section 2.2.3):

H1 The TUI would promote bimanual interaction, which would reduce the time needed to explore combinations of gene expressions and SNPs. This in turn would shorten the time needed to complete the task compared with the touch UI.

H2 The adoption of epistemic actions would result in more efficient data explorations using the TUI compared with the touch UI.

7.2.1 Participants

The study was approved by the Faculty Ethics Committee. All participants gave their informed consent and were paid for taking part in the experiment. Twenty individuals (8 men and 12 women) participated in the study with a mean age of 25.45 years (SD = 9.01). The participants were either students or employees at various universities and companies in the UK; one participant was a retired employee. The majority of participants (15 participants) were undergraduate or postgraduate students at the University of Leeds or Leeds Beckett University studying various disciplines (biology, mathematics, geography and psychology). Two participants were employees at the University of Leeds, two were employees at an insurance company and one was a retired teacher. All participants, with one exception, had little or no background in quantitative genetics. The participants were all right-handed and were familiar with touch technology from everyday use of tablets and smartphones.

7.2.2 Materials

The TUI's performance (see Section 6.5) was evaluated against a touch UI baseline (see Section 6.6). Table 7.1 shows the similarities between the tasks in our eQTL scenario and those identified in Brehmer and Munzner's wide-ranging review [22].

Generic Task	eQTL Task	Touch	Tangible
Encode	Open/close files	Open: drag and drop file from file explorer to the surface; Close: drag and drop file to recycle bin.	TAC 1
Change	Switch windows	Select window from the View menu.	TAC 2
Navigate	Scroll, pan and zoom	TAC 3; Scroll: Swipe up/down or move scroll bar; Pan: Swipe in the opposite direction of intended pan; Zoom: Pinch out/in to zoom in/out.	
Select	Select genetic variants	TAC 4; Select data point in plot or row in table.	
Arrange	Organise windows	Drag window on surface	TAC 6.1
Filter/Change	Filter data	Select filtering from the Threshold menu and rotate dial.	TAC 7
Aggregate	Combine files	Select file from the Group menu.	TAC 8
Derive	Match significance across files	Select file from the Threshold menu.	TAC 9

Table 7.1: The generic visualisation tasks [22] and the tangible (as TAC references from Table 6.8) and touch interactions utilised for each eQTL task.

The table also maps the tangible interactions from Table 6.8 and the multi-touch interactions to the generic visualisation tasks. For the purpose of this experiment, access to external data sources was disabled in both interfaces to focus exploration into combinations of gene expressions. Similar to the experiment carried out previously (see Section 4.4), combinations were limited to five gene expressions per combination.

7.2.2.1 Experimental Data Set

The Biovis 2012 [103] biological domain was eQTL association mapping, where a collection of genotype and gene expression data were provided to identify genetic variants of regulatory significance. The eQTL data set was analysed using PLINK [190], which resulted in association files for each gene expression. In each gene expression file, the 230,912 genetic variants were given statistical values to indicate the genetic variants' significance to the gene expression. A bin-width optimisation technique was used to formulate histograms of the resulting analysis files [218]. The analysis results and histogram files were then used as input to a small Java programme that produced fabricated data for this experiment.

Once new data sets were produced, gene expression files were assigned to tasks and intersecting variants between the files were introduced manually. The complexity of the data sets were determined by controlling the number of intersecting significant variants, i.e. the fewer the variants shared the fewer the combinations worth exploring. This level of control was used to encourage guided explorations and the utilisation of epistemic actions and bimanual interactions. Based on the results of a previous experiment (see Section 4.4) the number of items presented in the file explorer were kept below ten to reduce error.

To demonstrate the interfaces, six gene expression files were loaded: CNTN1, CNTN2, CNTN3, CNTN4, CNTN5 and CNTN6. Table 7.2 shows all possible explorations for the rest of the tasks (training, practice, task 1 and task 2) and the number of significant variants.

Task	Exploration	Number of SNPs
Training	CNTN1	10
	CNTN2	10
	CNTN3	9
	CNTN1, CNTN2	4
	CNTN1, CNTN3	0
	CNTN2, CNTN3	3
	CNTN1, CNTN2, CNTN3	0
Practice	AGRN	8
	CNTN1	9

Continued on next page

Task	Exploration	Number of SNPs
Practice	CNTN3	7
	CNTN4	12
	AGRN, CNTN1	2
	AGRN, CNTN3	3
	AGRN, CNTN4	2
	CNTN1, CNTN3	3
	CNTN1, CNTN4	5
	CNTN3, CNTN4	2
	AGRN, CNTN1, CNTN3	1
	AGRN, CNTN1, CNTN4	1
	AGRN, CNTN3, CNTN4	0
	CNTN1, CNTN3, CNTN4	2
	AGRN, CNTN1, CNTN3, CNTN4	0
Task 1	CNTN1	9
	CNTN2	11
	CNTN3	6
	CNTN5	9
	CNTN6	9
	CNTN1, CNTN2	1
	CNTN1, CNTN3	1
	CNTN1, CNTN5	2
	CNTN1, CNTN6	2
	CNTN2, CNTN3	2
	CNTN2, CNTN5	2
	CNTN2, CNTN6	0
	CNTN3, CNTN5	1
	CNTN3, CNTN6	0
	CNTN5, CNTN6	4
	CNTN1, CNTN2, CNTN3	0
	CNTN1, CNTN2, CNTN5	0
	CNTN1, CNTN2, CNTN6	0
	CNTN1, CNTN3, CNTN5	0
	CNTN1, CNTN3, CNTN6	0
	CNTN1, CNTN5, CNTN6	2

Continued on next page

Task	Exploration	Number of SNPs
Task 1	CNTN2, CNTN3, CNTN5	0
	CNTN2, CNTN3, CNTN6	0
	CNTN2, CNTN5, CNTN6	0
	CNTN3, CNTN5, CNTN6	0
	CNTN1, CNTN2, CNTN3, CNTN5	0
	CNTN1, CNTN2, CNTN3, CNTN6	0
	CNTN1, CNTN2, CNTN5, CNTN6	0
	CNTN1, CNTN3, CNTN5, CNTN6	0
	CNTN2, CNTN3, CNTN5, CNTN6	0
	CNTN1, CNTN2, CNTN3, CNTN5, CNTN6	0
Task 2	AGRN	8
	CNTN1	10
	CNTN2	6
	CNTN5	6
	CNTNAP1	5
	CNTNAP2	6
	AGRN, CNTN1	0
	AGRN, CNTN2	0
	AGRN, CNTN5	0
	AGRN, CNTNAP1	0
	AGRN, CNTNAP2	0
	CNTN1, CNTN2	0
	CNTN1, CNTN5	3
	CNTN1, CNTNAP1	0
	CNTN1, CNTNAP2	3
	CNTN2, CNTN5	0
	CNTN2, CNTNAP1	0
	CNTN2, CNTNAP2	2
	CNTN5, CNTNAP1	0
	CNTN5, CNTNAP2	2
	CNTNAP1, CNTNAP2	0
	AGRN, CNTN1, CNTN2	0
	AGRN, CNTN1, CNTN5	0
	AGRN, CNTN1, CNTNAP1	0

Continued on next page

Task	Exploration	Number of SNPs
Task 2	AGRN, CNTN1, CNTNAP2	0
	AGRN, CNTN2, CNTN5	0
	AGRN, CNTN2, CNTNAP1	0
	AGRN, CNTN2, CNTNAP2	0
	AGRN, CNTN5, CNTNAP1	0
	AGRN, CNTN5, CNTNAP2	0
	AGRN, CNTNAP1, CNTNAP2	0
	CNTN1, CNTN2, CNTN5	0
	CNTN1, CNTN2, CNTNAP1	0
	CNTN1, CNTN2, CNTNAP2	0
	CNTN1, CNTN5, CNTNAP1	0
	CNTN1, CNTN5, CNTNAP2	2
	CNTN1, CNTNAP1, CNTNAP2	0
	CNTN2, CNTN5, CNTNAP1	0
	CNTN2, CNTN5, CNTNAP2	0
	CNTN2, CNTNAP1, CNTNAP2	0
	CNTN5, CNTNAP1, CNTNAP2	0
	AGRN, CNTN1, CNTN2, CNTN5	0
	AGRN, CNTN1, CNTN2, CNTNAP1	0
	AGRN, CNTN1, CNTN2, CNTNAP2	0
	AGRN, CNTN1, CNTN5, CNTNAP1	0
	AGRN, CNTN1, CNTN5, CNTNAP2	0
	AGRN, CNTN1, CNTNAP1, CNTNAP2	0
	AGRN, CNTN2, CNTN5, CNTNAP1	0
	AGRN, CNTN2, CNTN5, CNTNAP2	0
	AGRN, CNTN2, CNTNAP1, CNTNAP2	0
	AGRN, CNTN5, CNTNAP1, CNTNAP2	0
	CNTN1, CNTN2, CNTN5, CNTNAP1	0
	CNTN1, CNTN2, CNTN5, CNTNAP2	0
	CNTN1, CNTN2, CNTNAP1, CNTNAP2	0
	CNTN1, CNTN5, CNTNAP1, CNTNAP2	0
	CNTN2, CNTN5, CNTNAP1, CNTNAP2	0
	AGRN, CNTN1, CNTN2, CNTN5, CNTNAP1	0
	AGRN, CNTN1, CNTN2, CNTN5, CNTNAP2	0

Continued on next page

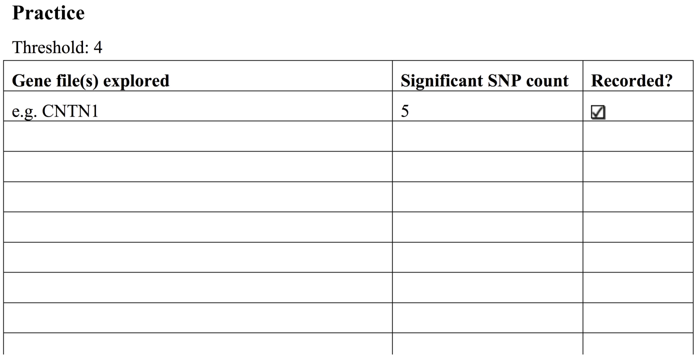
Task	Exploration	Number of SNPs
Task 2	AGRN, CNTN1, CNTN2, CNTNAP1, CNTNAP2	0
	AGRN, CNTN1, CNTN5, CNTNAP1, CNTNAP2	0
	AGRN, CNTN2, CNTN5, CNTNAP1, CNTNAP2	0
	CNTN1, CNTN2, CNTN5, CNTNAP1, CNTNAP2	0

Table 7.2: The various tasks' explorations and the number of significant SNPs for each exploration.

7.2.2.2 Experimental Task

Given a limited list of gene expression files, the participants were asked to explore the files separately and in combinations. With each exploration, they were also asked to record the number of SNPs that met a preset significance threshold ranging from $-log_{10}3$ to 5. This significance range was recommended in the Biovis 2012 competition [103] and at the threshold for each task was presented to the user in the written notes (see Figure 7.1) and verbally announced with the instructions.

In the case of exploring gene expression files individually, the participants were expected to open a file, display the file as a plot or table, adjust the threshold to collect SNPs that satisfy the significance threshold, display collected SNPs and record the number of SNPs in the collection. When exploring patterns between two or more files, the participants were expected to either combine the files or



Practice

Threshold: 4

Gene file(s) explored	Significant SNP count	Recorded?
e.g. CNTN1	5	☑

Figure 7.1: For each exploration, the participants kept written records using this sheet.

Task	Number of files					Total
	One	Two	Three	Four	Five	
Training	3	3(3)	1(0)	-	-	7(6)
Practice	4	6(6)	4(4)	1(0)	-	15(14)
Task 1	5	10(10)	10(5)	5(0)	1(0)	31(20)
Task 2	6	15(15)	20(1)	15(0)	6(0)	62(22)

Table 7.3: The total number of combinations that needs to be checked for each task. The minimum number of combinations that needs to be checked is shown in parentheses.

match their significance. If the participants decided to combine files, they were expected to open two or more gene expression files, use one file as a base and start combining files to it, adjust the threshold to collect SNPs that satisfy the significance threshold in all files in the combination, display the collected SNPs and record the number of SNPs in the collection. When adopting the match approach, participants were expected to open two or more gene expression files, adjust the threshold to collect SNPs that satisfy the significance threshold in one file and match their significance against other files.

After each gene expression file or combination was explored, the participants were asked to record their results in two ways: a written record that listed the files' names and the number of significant SNPs that met the threshold (see Figure 7.1) and a digital record captured by pressing the record button anchored to the corner of the display (see Figure 6.19).

The number of combinations that the participants needed to explore was dependent on the number of files in a task and the strategies used to form combinations and explore comparisons. For example, if two files did not share any significant SNPs, then there was clearly no need to check any other combinations that included those two files. Table 7.3 shows the total number of combinations in the various tasks and the minimum number of combinations that needed to be checked.

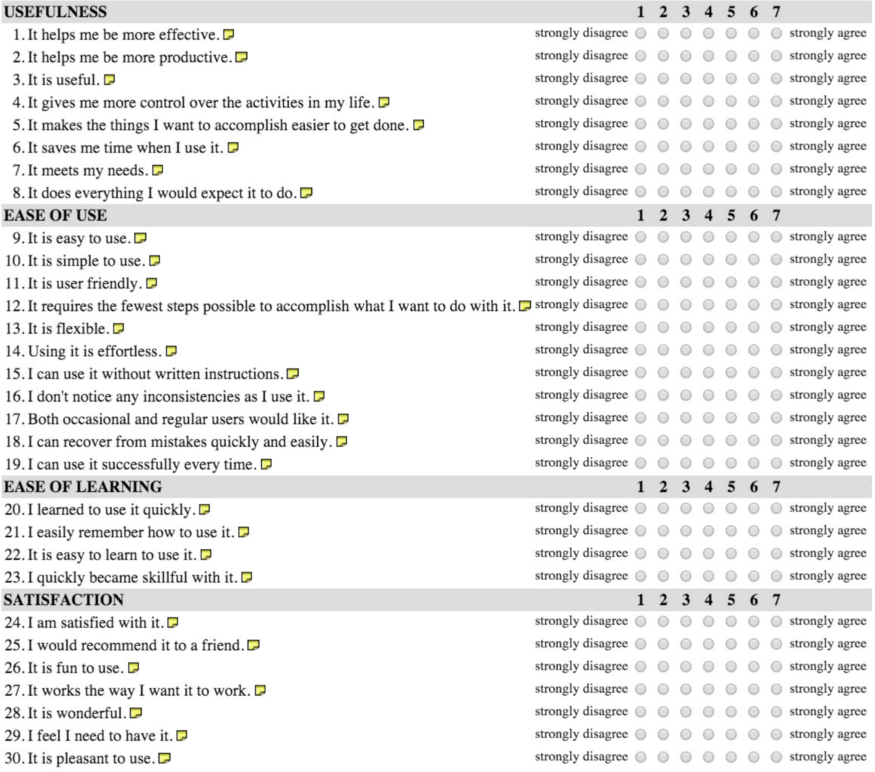
7.2.3 Procedure

Each participant was randomly assigned to one of the two experimental conditions (tangible versus touch). Sessions were held in a quiet laboratory space and each session lasted around 90 min.

At the start of the session, a participant was given a demonstration on using the interface. The demonstration data set consisted of six files: CNTN1, CNTN2, CNTN3, CNTN4, CNTN5 and CNTN6. The demonstration was carried out by the experimenter as follows:

1. Scroll through the file explorer to expose all gene expression files.
2. Open CNTN1 gene expression file and explain CNTN1's default window (Manhattan plot).
3. Zoom and pan plot.
4. Tap on data point to expose additional information (infotip) about the SNP.
5. Reset plot's view.
6. Switch windows from plot to table and explain CNTN1's table window.
7. Scroll table.
8. Tap on row to highlight row.
9. Hide or minimise window.
10. Maximise window and display as plot.
11. Adjust threshold to $-log_{10}3$ and collect SNPs that meet the requirement.
12. Display collected SNPs.
13. Open CNTN4 gene expression file.
14. Match the significance of collected SNPs (step 11) against CNTN4's default window. Explain the interaction and record the number of SNPs that match.
15. Combine CNTN1 and CNTN4 and perform steps 11 and 12 on the combination.
16. Compare the number of collected SNPs with the results from step 14 (the values should match) and explain the interaction.
17. Discard of CNTN1 and CNTN4 files.

After a two minute break (the participants were encouraged to interact with the system during that break) the task instructions were verbalised and the threshold pointed out in the written sheet. The training task consisted of three



USEFULNESS		1	2	3	4	5	6	7	
1. It helps me be more effective.	strongly disagree	○	○	○	○	○	○	○	strongly agree
2. It helps me be more productive.	strongly disagree	○	○	○	○	○	○	○	strongly agree
3. It is useful.	strongly disagree	○	○	○	○	○	○	○	strongly agree
4. It gives me more control over the activities in my life.	strongly disagree	○	○	○	○	○	○	○	strongly agree
5. It makes the things I want to accomplish easier to get done.	strongly disagree	○	○	○	○	○	○	○	strongly agree
6. It saves me time when I use it.	strongly disagree	○	○	○	○	○	○	○	strongly agree
7. It meets my needs.	strongly disagree	○	○	○	○	○	○	○	strongly agree
8. It does everything I would expect it to do.	strongly disagree	○	○	○	○	○	○	○	strongly agree
EASE OF USE		**1**	**2**	**3**	**4**	**5**	**6**	**7**	
9. It is easy to use.	strongly disagree	○	○	○	○	○	○	○	strongly agree
10. It is simple to use.	strongly disagree	○	○	○	○	○	○	○	strongly agree
11. It is user friendly.	strongly disagree	○	○	○	○	○	○	○	strongly agree
12. It requires the fewest steps possible to accomplish what I want to do with it.	strongly disagree	○	○	○	○	○	○	○	strongly agree
13. It is flexible.	strongly disagree	○	○	○	○	○	○	○	strongly agree
14. Using it is effortless.	strongly disagree	○	○	○	○	○	○	○	strongly agree
15. I can use it without written instructions.	strongly disagree	○	○	○	○	○	○	○	strongly agree
16. I don't notice any inconsistencies as I use it.	strongly disagree	○	○	○	○	○	○	○	strongly agree
17. Both occasional and regular users would like it.	strongly disagree	○	○	○	○	○	○	○	strongly agree
18. I can recover from mistakes quickly and easily.	strongly disagree	○	○	○	○	○	○	○	strongly agree
19. I can use it successfully every time.	strongly disagree	○	○	○	○	○	○	○	strongly agree
EASE OF LEARNING		**1**	**2**	**3**	**4**	**5**	**6**	**7**	
20. I learned to use it quickly.	strongly disagree	○	○	○	○	○	○	○	strongly agree
21. I easily remember how to use it.	strongly disagree	○	○	○	○	○	○	○	strongly agree
22. It is easy to learn to use it.	strongly disagree	○	○	○	○	○	○	○	strongly agree
23. I quickly became skillful with it.	strongly disagree	○	○	○	○	○	○	○	strongly agree
SATISFACTION		**1**	**2**	**3**	**4**	**5**	**6**	**7**	
24. I am satisfied with it.	strongly disagree	○	○	○	○	○	○	○	strongly agree
25. I would recommend it to a friend.	strongly disagree	○	○	○	○	○	○	○	strongly agree
26. It is fun to use.	strongly disagree	○	○	○	○	○	○	○	strongly agree
27. It works the way I want it to work.	strongly disagree	○	○	○	○	○	○	○	strongly agree
28. It is wonderful.	strongly disagree	○	○	○	○	○	○	○	strongly agree
29. I feel I need to have it.	strongly disagree	○	○	○	○	○	○	○	strongly agree
30. It is pleasant to use.	strongly disagree	○	○	○	○	○	○	○	strongly agree

Figure 7.2: The USE questionnaire [148].

files: CNTN1, CNTN2 and CNTN3, and the following steps were repeated for each training session:

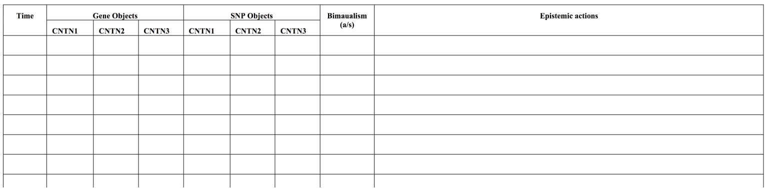
1. Open CNTN1 gene expression file.
2. Adjust threshold to $-log_{10}3$, collect SNPs that meet the requirement and display collected SNPs.
3. Record the file's name and number of collected SNPs in the table (see Figure 7.1). Tap on the record button to log the results to the system.
4. Participant repeats steps 1-3 with CNTN3.
5. Combine CNTN1 and CNTN3 gene expression files.
6. Adjust threshold to $-log_{10}3$, collect SNPs that meet the requirement in both files and display collected SNPs.
7. Record the files' names and the number of collected SNPs in the table (see Figure 7.1). Tap on the record button to log the results into the system. The combination (CNTN1 and CNTN3) results in zero significant SNPs.

8. Participant repeats steps 5-7 with CNTN1 and CNTN2 gene expression files. The combination results in four significant SNPs.
9. Combine CNTN1 and CNTN2 (from previous step) with CNTN3 to explain that if two files share no significant SNPs (CNTN1 and CNTN3), then there clearly was no need to check any other combinations that included those two files (CNTN1, CNTN2 and CNTN3).
10. Participant repeats steps 1-3 with CNTN1 and CNTN2 gene expression files.
11. Match the significance of collected SNPs from CNTN1 against CNTN2 and note the matches.
12. Participant inversely repeats step 11.

After training had been completed, the instructions were repeated for the practice task, task 1 and task 2. The participant took a two minute break between tasks. The experiment concluded with the participant completing the USE questionnaire [148], which gathered feedback about the usability of the system and its interface on a seven-point Likert scale (see Figure 7.2). This also included recording any positive or negative comments about the interface.

7.2.3.1 Observation Sheet

The participants' interactions were also recorded by an observer to track bimanual interaction and epistemic actions. For each exploration, the observer recorded bimanual interactions by specifying when it occurred and its type (asynchronous or synchronous). Epistemic actions were also tracked and included vocalisation, ordering, clustering or rearranging objects, hovering a hand over an object, pointing, fiddling with an object and dividing the interactive surface. Figure 7.3 shows the observation sheet used to record the participants' interactions.



Time	Gene Objects			SNP Objects			Bimaualism (a/s)	Epistemic actions
	CNTN1	CNTN2	CNTN3	CNTN1	CNTN2	CNTN3		

Figure 7.3: Observation sheet used to track bimanual interaction and epistemic actions.

7.2.3.2 Logging Participants' Activities

Participants' interactions with the interface were recorded using a logger class linked to both interfaces. Combination results were also logged when the participant used the record button to save their results. Table 7.4 summarises the information logged when interacting with the interfaces and the exploration results that were recorded by the participants.

Logged value	Description
Duration	Session duration
Type of interaction	Touch or tangible
Object type	Gene expression or SNP
Action	Gene expression (add to surface, move on surface, remove from surface, add to combination, remove from combination), SNP (add to window, move on window, remove from window, add to surface, move on surface, remove from surface), touch (down, up)
Tag value	Long integer tag value
Object function	Gene expression (table, plot), SNP (collection, dial)
Gene expression file	Gene expression file's name
Position	x and y position on the surface
Orientation	Orientation on the surface
Exploration duration	Duration of each exploration
Gene expression files	The names of the gene expression files in a combination
Significance threshold	Significance threshold value
Number of significant SNPs	The number of collected significant SNPs
Significant SNPs	The list of collected significant SNPs
Print screen	A print screen of each exploration

Table 7.4: Logged values and descriptions.

7.3 Results

The evaluation results are presented in the following three subsections. First, the log files are analysed to explore the objective metrics: session time, exploration time and number of combinations explored. Next, the subjective metrics relating to strategies and inefficiencies are examined. Finally, the results of the USE questionnaire and participants' comments are described.

7.3.1 Objective Metrics

The results were analysed using mixed factorial ANOVA that treated the interface as a between-participants factor (tangible versus touch) and the task as a repeated measure (task 1 versus task 2).

Overall performance was measured by calculating the total time that the participants took to complete each task. An ANOVA showed that task completion was significantly faster with the TUI than with the touch UI ($F_{1,18} = 6.64, p = .02$) and was significantly faster for task 1 than task 2 ($F_{1,18} = 14.89, p < .01$). There was also a significant interface × task interaction ($F_{1,18} = 10.74, p < .01$), with

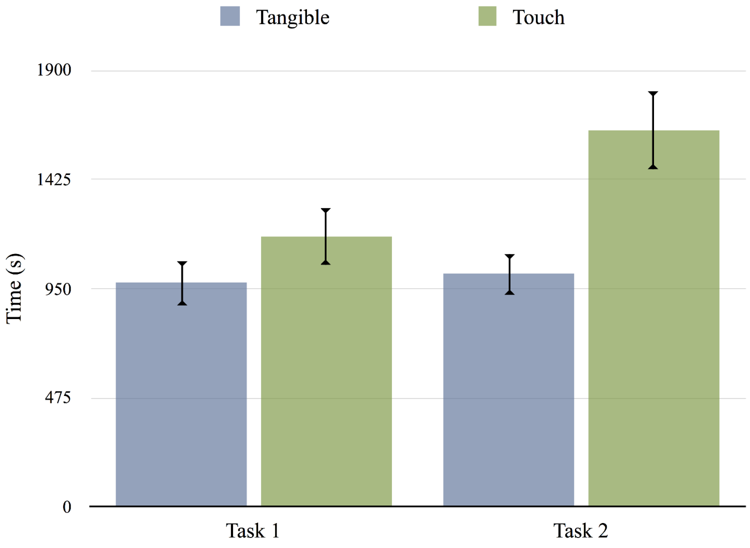


Figure 7.4: Mean task completion time for each condition and task. Error bars show the standard error of the mean (SE).

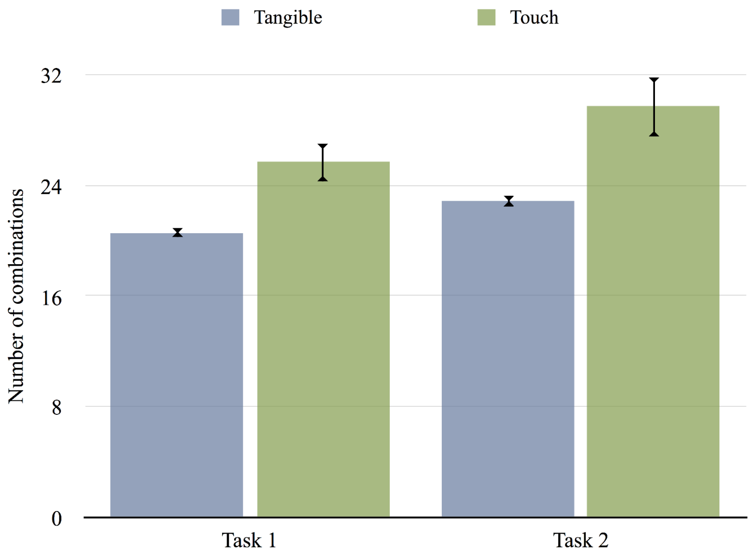


Figure 7.5: Mean number of combinations checked for each condition and task. Error bars show the standard error of the mean (SE).

the time difference between the two interfaces increasing with the number of files involved in the task (see Figure 7.4).

To investigate the overall performance difference, two separate analyses were performed. First, the time that participants took to check each exploration was gathered from the log file data (exploration duration). An ANOVA showed that there was no significant difference between the conditions ($F_{1,18} = .35, p = .56$) or tasks ($F_{1,18} = .06, p = .81$). Second, the number of combinations that participants checked in each task was analysed. An ANOVA showed that they checked fewer combinations with the TUI than with the touch UI ($F_{1,18} = 13.93, p < .01$). Furthermore, fewer files were explored in task 1 than task 2 ($F_{1,18} = 11.89, p < .01$). See Figure 7.5.

7.3.2 Subjective Metrics

The experimental task can be represented as a tree that the participants needed to traverse: the root is the start of the task, the individual files are at level 1, the combinations involving two files are at level 2 and so on. Three strategies were

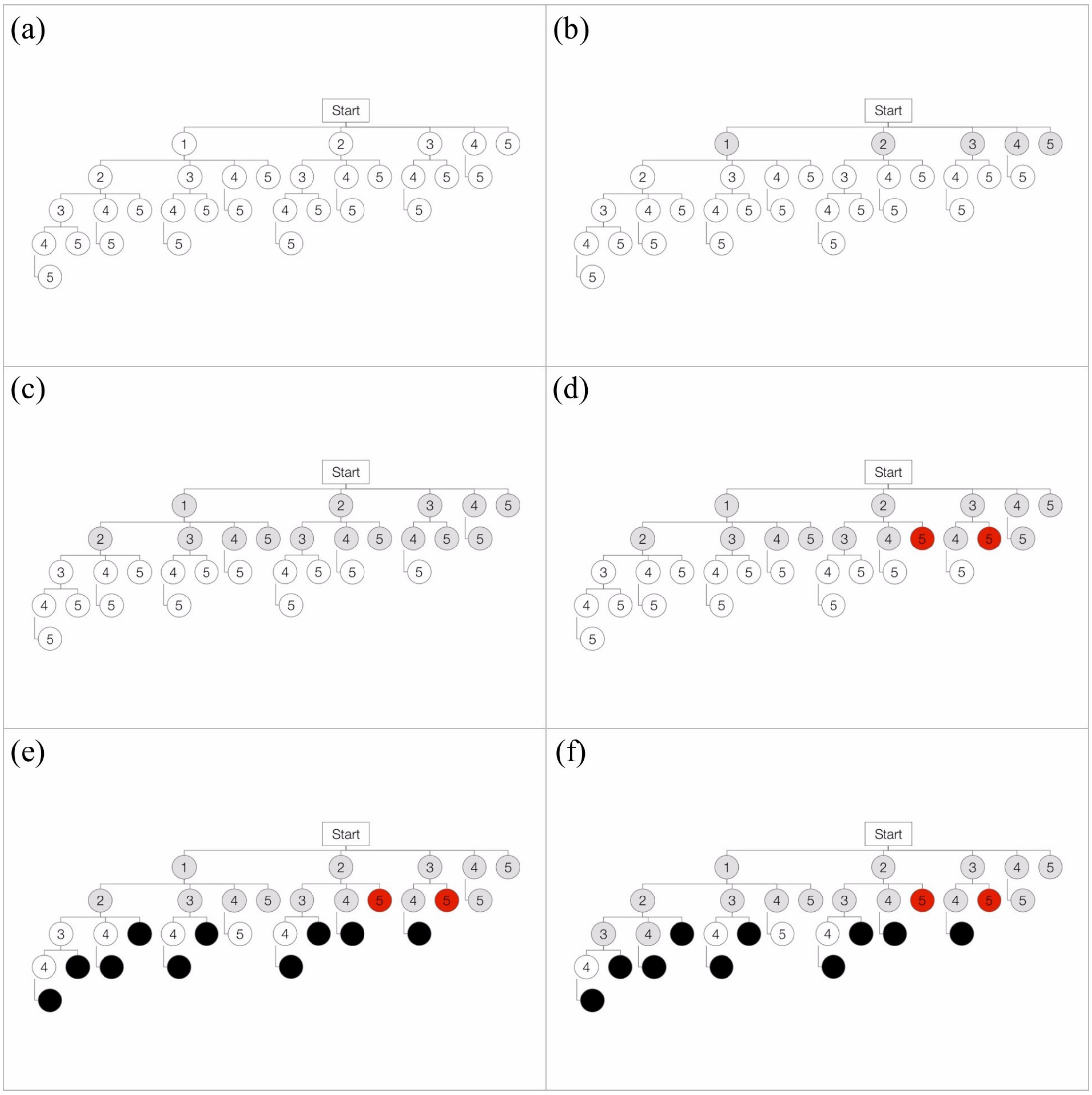


Figure 7.6: Breadth-first traversal strategy. (a) The task is represented as a tree, where each node is a gene expression file. (b) All files are explored separately on the first level. (c) All two file combinations are explored next on the second level. (d) Combinations resulting in no shared significant SNPs are highlighted in red. (e) The results from the previous level's combinations - red nodes - means it is impossible for some of the following levels' combination to have any shared significant SNPs - black nodes. (f) Continue exploring the next level.

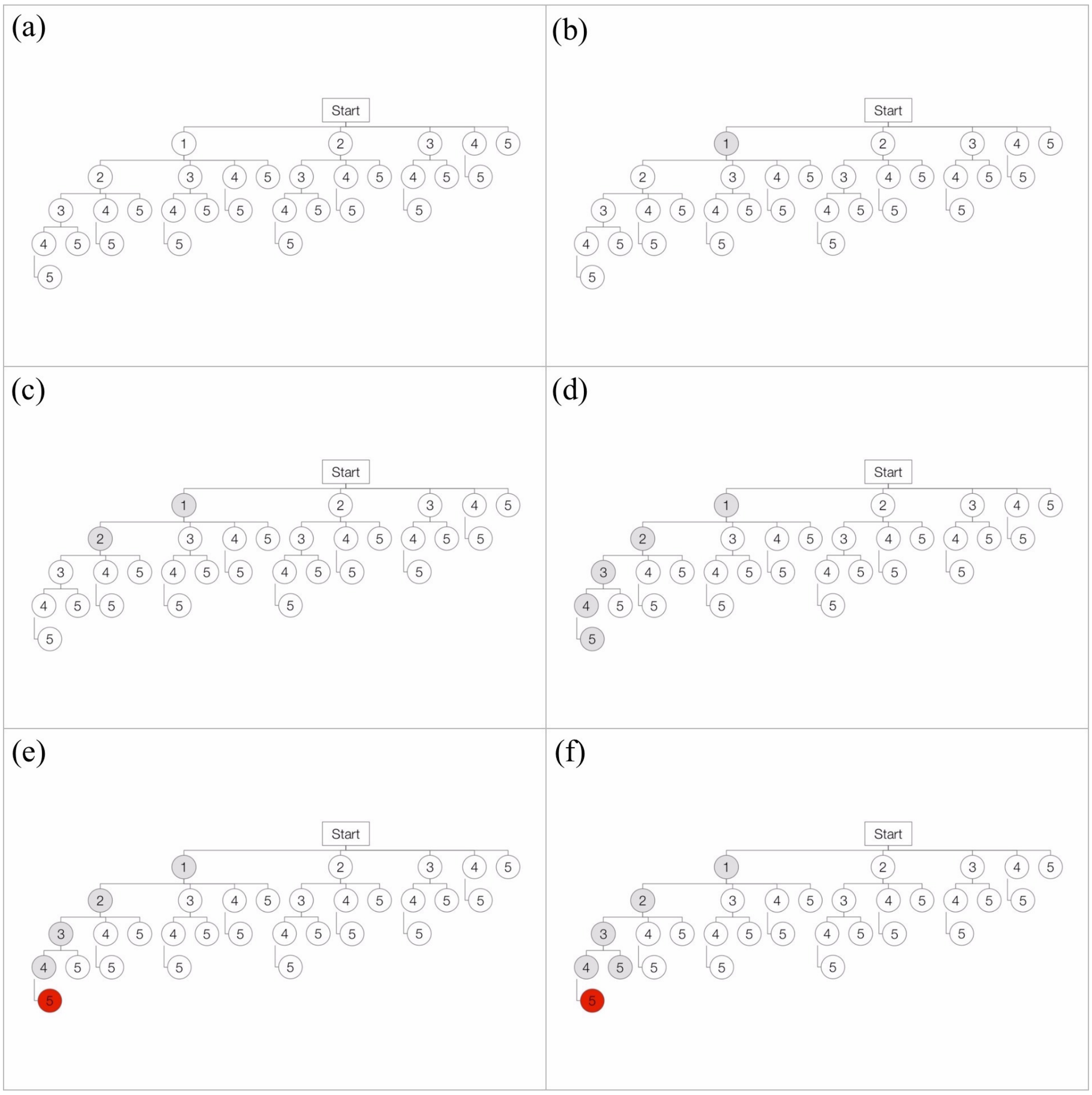


Figure 7.7: Depth-first traversal strategy. (a) The task is represented as a tree, where each node is a gene expression file. (b) A single file is explored in the first level. (c) A single two files combination is explored on the second level. (d) A single combination is explored in each level until a leaf is reached. (e) Combinations resulting in no shared significant SNPs are highlighted in red. (f) Reverse one level to reach another leaf combination.

adopted by the participants to traverse the tree: breadth-first, depth-first and mixed.

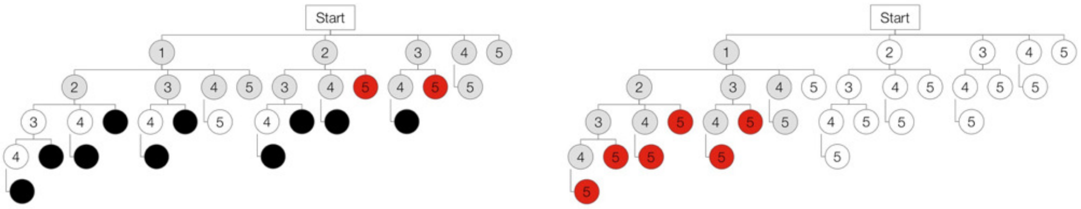


Figure 7.8: Breadth-first versus depth-first traversal for task 1 after 15 explorations (gene expression files are numbered in the order they appear in Table 7.2). Red nodes are explorations resulting in no shared significant SNPs. Black nodes are eliminated and not explored since a previous combination meant that it was impossible for the present combination to have any shared significant SNPs.

A breadth-first traversal went through combinations level-by-level and started by checking all combinations that involved pairs of files (see Figure 7.6). A depth-first traversal completed checks for a given branch of the tree (e.g. all combinations that involve two particular files) before it explored sibling combinations (see Figure 7.7). The mixed approach traversed the tree two levels at a time, combining the breadth-first and depth-first strategies. The breadth-first strategy allowed participants to eliminate the most file combinations and speeded up the analysis (see Figure 7.8).

Most participants in the tangible condition used breadth-first traversal for both tasks, whereby in the touch condition they adopted all three strategies (see Table 7.5). Most participants used the same strategy for both tasks.

Three sources of inefficiency were noted:

- Repetition: checking the same combination of gene expression files more than once.
- Unnecessary: the results of a previous combination meant that it was impossible for the present combination to have any shared significant SNPs.
- Strategy: it would not have been necessary to check the combination if a more effective strategy had been adopted.

Unnecessary and strategy inefficiencies accounted for most of the additional analyses that participants performed with the touch UI (see Table 7.6).

Bimanual interactions were observed and recorded for both tangible and touch interactions. For each exploration, the observer noted how many times bimanulism was used and its type. For the touch UI, bimanual interactions were utilised 9.04% and 7.83% of the time for tasks 1 and 2 respectively. TUI participants used both hands 11.54% and 10.79% of the time for tasks 1 and 2. Two-handed interactions for both conditions were largely categorised as either asynchronous or concurrent unimanualism [240].

7.3.3 USE Questionnaire and Comments

The USE questionnaire data indicated that the participants found the TUI to be slightly more useful, easier to use, easier to learn and satisfactory. However, the touch UI was rated slightly higher for the time it took to learn and its simplicity. Subsequent analysis with Bonferroni corrected t-tests showed that none of the differences were significant.

Condition	Task	Breadth-first	Depth-first	Mixed
Tangible	Task 1	9	0	1
	Task 2	10	0	0
Touch	Task 1	3	4	3
	Task 2	3	3	4

Table 7.5: Number of times each strategy was used by participants for each condition and task.

Condition	Task	Repetition	Unnecessary	Strategy
Tangible	Task 1	0	0.4	0.2
	Task 2	0.1	0.8	0
Touch	Task 1	0.7	2.6	2.4
	Task 2	0.5	3.9	3.4

Table 7.6: Mean number of inefficiencies that occurred in a trial for each condition and task.

The comments from the USE questionnaire were analysed to summarise the participants' experience using the interfaces. Almost half of the participants stated that the touch UI was easy to use and learn, with one participant saying 'It certainly is more enjoyable to use than a regular computer - like something from a science fiction movie!'. Another participant remarked that it was easier to navigate than a keyboard. Two participants found the display's size to be a positive feature. One participant declared that the tabletop concept and the physical movements required to interact with the various features were an added bonus.

Some of the touch UI participants made comments about the touch response. One participant particularly found the touch slow to respond, 'Can be a little infuriating to use at times, particularly when it is not doing exactly what you want it to'. Another thought that the touch aspect was easy to use but not as fast as a keyboard. Suggestions were also given by some of the participants about how to improve the tool. One participant thought that some tasks could have been better automated.

For the TUI, half of the participants highlighted the interface's user friendliness and ease of use. Two participants also found the interface fun to use, and in one case, 'better than a touchscreen'. The term interactive was used by two participants to describe their interactions with the TUI. Other participants made comments about the display, some finding it 'clear and bright'. Some of the participants also made positive comments about the tangible objects:

> 'Can grasp table/graphs in a more practical format'
> 'Easy to pick up'
> 'I like the 3D aspect compared to a similar touchscreen'

> ‘Easy to use - one tap to combine data’

Two of the participants thought that the display was too bright and one found it uncomfortable to stare at the screen for too long. One participant believed the TUI will be difficult to learn by the older generation, stating ‘Even though it was pretty intuitive, the older generation may consider it a new and different experience. There are slight challenges in learning’. Another participant remarked that the use of base gene expression objects hindered the exploration of combinations. It would be possible to overcome this with the use of dedicated combination gene expression objects that were explored in the design options (see Section 5.3).

The participants also gave useful comments about improving the tool. One participant suggested the use of shortcuts, and this might be possible with the use of more specific and dedicated objects. Two participants found that having to set the same threshold for various combinations tiresome. They suggested the use of an object (possibly a SNP object) where the threshold value would be set once. The object should then be used to automatically reset a threshold as soon as it is placed on a gene expression file’s window, eliminating the need to turn the dial to adjust.

7.4 Discussion

The participants completed the interactive visualisation task significantly faster with the TUI than with the touch UI as they explored combinations more effectively. This section discusses these findings in the context of our hypotheses regarding bimanual interaction (**H1**) and epistemic actions (**H2**).

A few participants took advantage of the bimanual capability of the multi-touch and tangible interactions; this is consistent with the previous findings for tangible [234, 235, 240] and touch interactions [8, 234, 235]. When the participants utilised bimanual interaction in the TUI, the interactions were asymmetrical or involved one hand moving objects out of the way while simultaneously opening or filtering a file with a new object, i.e. both hands are working separately on independent tasks. The latter was previously described as concurrent unimanualism [240]. These results support previous findings [235, 240]. There was no evidence to support hypothesis **H1**, which postulated that the TUI would encourage bimanual interaction more often than the touch UI and reduce the time it took to analyse file combinations.

Cognitive processes are augmented by epistemic actions which facilitate and reduce the need for internal computations. In a direct manipulation interface, users can see and manipulate virtual content directly. Tangible and touch UIs inherently matches users' interactions and expectations to the system's interpretation. These types of interface also reduce the gulf between intended actions and affordance. Compared to a conventional interface with a mouse and keyboard, users can closely map their intentions into touch and TUI systems. In this experiment the TUI and touch UI were compared against each other to determine how the tasks were simplified using epistemic actions.

Findings from previous studies suggest that TUIs promote epistemic actions that encourages more effective and efficient motor-cognitive strategies to solve tasks [7, 8, 157, 183]. We hypothesised (**H2**) that the adoption of epistemic actions would result in more efficient exploration during data visualisation with the TUI rather than with the touch UI. The results supported **H2** as participants explored combinations more efficiently. Repetitions and unnecessary explorations were reduced and more effective strategies were adopted. This in turn reduced the time spent on a task with the TUI compared with the touch UI.

One of the subjective metrics addressed the strategies adopted by users when exploring combinations. Of the three identified strategies (breadth-first, depth-first, and mixed traversals) breadth-first was regularly adopted by the majority of users in the TUI condition, while the touch UI's participants equally utilised breadth-first, depth-first, and mixed traversals. Assuming a tree structure to traverse a breadth-first strategy goes through combination level-by-level (see Figure 7.6), whereby a depth-first traversal completed combinations for a given branch (see Figure 7.7). Mixed traversals combines breadth-first and depth-first approaches by going through combinations two levels at a time. Of the three strategies, breadth-first traversal aids participants in eliminating unsuccessful combinations early on the task.

When using breadth-first traversals in the TUI condition, participants coupled all gene expression files with respective objects and aligned them somewhere around the screen (the files were minimised so as not to take too much room). Coupling the files with the objects in the start of the task simplified the process of exploring combinations. This alignment of the objects once they were coupled supported the breadth-first traversal and aided the user in approaching the traversal one level at a time. For instance, one file was opened and then combination were formed one object at a time until the next level. Explorations resulting in no significant genetic variants were eliminated as early as the second level, and because of the level-by-level approach of this strategy users were able to formulate physical groups that can potentially share significant genetic variants. These groups, along with the written records, guided the participants in the exploration of higher levels. The alignment and grouping of objects were interpreted as epistemic actions to facilitate explorations.

In the touch UI condition, a third of the participants adopted breadth-first traversals to explore combinations. In this case, a participant typically opened one file and proceeded to explore two file combinations using that file as a base. Once the two-file explorations were completed for the first file, the participant opened the next file and explored them in a similar manner. In all cases, the files were minimised and the windows hidden from view once the exploration of that level was completed. The nature of the system allowed for the ring identifiers to overlap and the majority of participants allowed for the overlap to occur (see Figure 7.9). It can be argued that the varying behaviour between participants of the touch UI and TUI is due to the physical barrier that the tangible objects enforce. When exploring higher level (e.g. 3 level combinations) participants in the touch UI condition heavily relied on the results recorded on the written notes to either preemptively determine which combination to explore at this juncture or during their explorations.

Depth-first traversal was not used by TUI participants, but a third of the touch UI participants adopted the strategy for their explorations. In this strategy, a participant opened one gene expression file and explored all the levels stemming from that file. Combinations in the touch UI are explored by using a desktop idiom, i.e. menu control to add and remove files to/from a combination. This design did not require participants to open another file to form a combination (i.e. combinations can be created autonomously). While only a third of the participants adopted a depth-first traversal, the use of the desktop idiom might have contributed to the adoption of this strategy. This opens up a future opportunity to explore

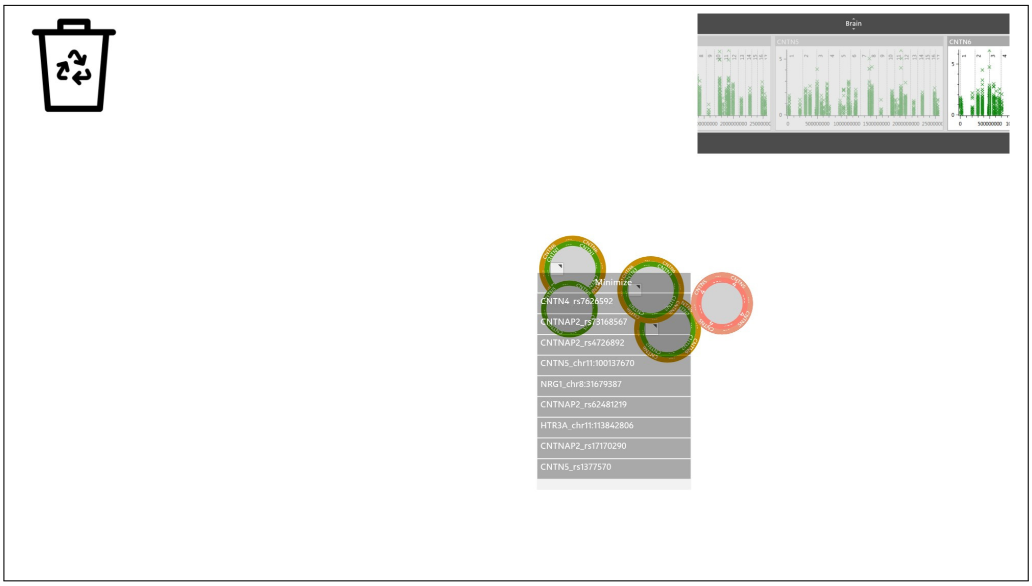


Figure 7.9: The identifier rings overlapping in the touch UI.

the effects of using desktop idiom and gestures in a touch UI on the utilisation of epistemic actions.

The full artefact, tool and body (ATB) framework is a recently developed video-coding framework that enables the identification and measurement of different epistemic actions during problem-solving tasks [57]. The ATB framework divides epistemic actions into three groups on the basis of whether the actions are performed with task artefacts, tools or users' own bodies (see Table 7.7). Since sessions were not video recorded, the ATB framework could not be adopted as a video-coding framework but types of epistemic actions presented in the framework were used to identify epistemic actions in combination with screenshots of the participants' results and the observations. While epistemic actions were noted for the TUI for the majority of participants, it was not the same case for the touch UI. The remainder of this discussion reports on the epistemic actions performed with the TUI.

The majority of the participants spatially arranged artefacts (ATB A2) in relation to one another or the task environment. For example, in both tasks, 8 of the 10 participants spatially arranged gene expression objects following the order they appear in the file explorer and maintained that order as the combinations were explored (see Figure 7.10). This seems to aid the strategy adopted when they are

ATB	Epistemic actions
A1	Manipulation of an artefact
A2	Spatial arrangement of artefacts in relation to one another, the task environment, or the users
2.1	Cluster or group artefacts together
2.2	Divide workspace into several stations in which only a subset of actions are afforded
2.3	Place an artefact in contrasting environment
2.4	Rearrange a representation
2.5	Clear and clean clutter
A3	Parallel use of two artefacts, two representations, or an artefact and a representation
A4	Artefact trial-and-error positioning
A5	Shuffle artefacts
A6	Compare an artefact with a possible destination or other artefacts
A7	Mark an artefact
A8	Test the state or response of a system, model or other user
Manipulation of a tool	
T9	Tag or annotate an artefact
T10	General notes and annotations
T11	Use of a tool to physically constraint the user or the use of other artefacts and tools
T12	Build a model or external representation
Bodily action	
B13	Use the body to externalise an internal process
B14	Talk or gesture to guide and direct attention
B15	Move the body, problem space, or representation

Table 7.7: List of all 20 epistemic actions present in the ATB framework [57]. Actions observed in the TUI condition of the experiment are highlighted in yellow.

exploring combinations at the earliest stage of their exploration of combinations. This early adoption of epistemic actions was previously reported [7].

As the participants eliminated combinations that did not have significant SNPs, they clustered or grouped gene expression objects together (ATB 2.1) to indicate

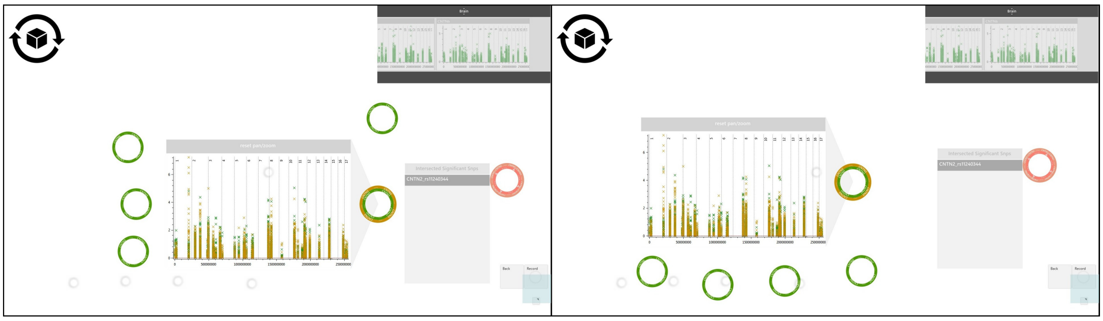
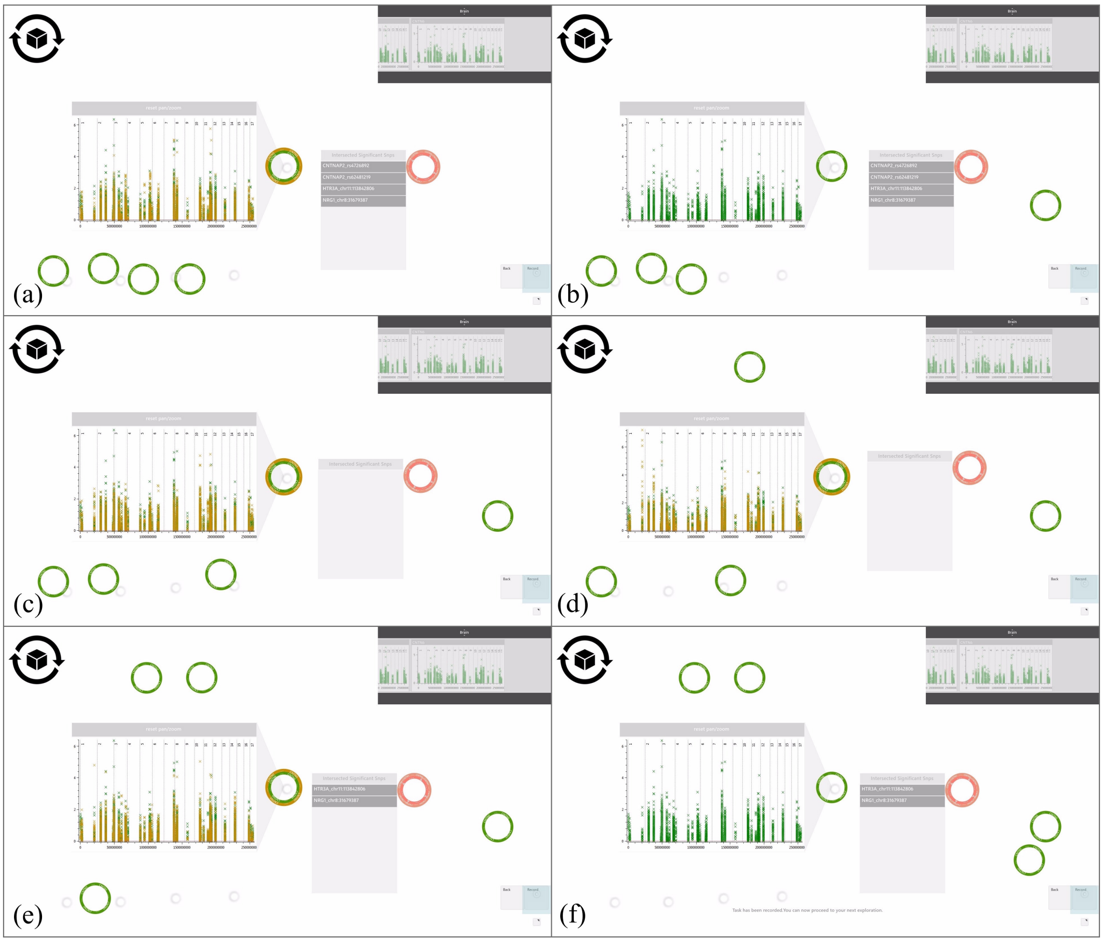


Figure 7.10: Gene expression objects spatially arranged by two different participants to aid exploration.

possibly successful combinations (see Figure 7.11). This was also observed for all except one participant. This action clearly externalises the internal process of forming combinations (ATB B13). The participants were also found to divide the surface into various stations in which certain actions are afforded (ATB 2.2). Figure 7.12 shows a participant's display division. The base gene expression window and SNP collection are displayed in the centre of the surface. Gene expressions still to explore are kept below the base window, while those already explored are placed in the area above the window. Gene expression files that shared significance with the base file are clustered to the right.

Objects were often rearranged and shuffled (ATB 2.4 and A5) by half of the participants to adhere to the order the participants established for them at the beginning of their exploration. Objects were repeatedly cleared and cleaned (ATB 2.5) by utilising the object recycler or placing the objects on the borders of the tabletop. On one occasion, the object recycler was used as a dump for unused objects that would need to be recycled but not instantly needed. This action was observed and recorded by the experimenter, where the objects were clustered in that area for recycling. When objects were placed on the border, most participants tested the state of that object (ATB A8), i.e. the file it contained, prior to deciding on its placements (ATB A4).

As was noted earlier, only a few participants took advantage of the bimanual capabilities of the touch UI and TUI. When bimanual interaction was observed, it involved the parallel use of two artefacts for two different tasks (ATB A3), for example, one hand moving an object out of the way while simultaneously opening



Intersected Significant Snps

CNTNAP2_rs4726892

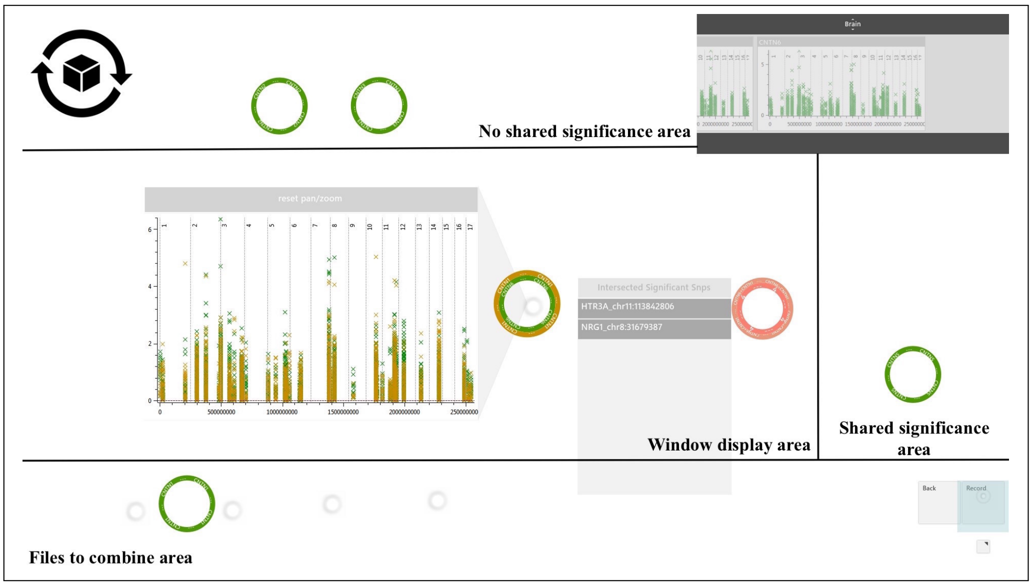
CNTNAP2_rs62481219

HTR3A_chr11:113842806

NRG1_chr8:31679387

Task has been recorded.You can now proceed to your next exploration.

Figure 7.11: As the exploration progresses, the participant started forming groups of objects that meet certain conditions that are kept in designated areas. (a) The base gene expression object, CNTN6, is combined with CNTN5 and results in four shared significant SNPs. (b) CNTN5 is removed from the combination and moved to a new area. (c) CNTN6 and CNTN3 are combined and no significant SNPs are shared. (d) CNTN3 is removed from the combination and moved to the area above the base window. CNTN6 and CNTN2 are combined and similarly result in no shared significant SNPs. (e) CNTN2 is removed from the combination and moved next to CNTN3. CNTN6 and CNTN1 are combined and result in two shared significant SNPs. (f) CNTN1 is removed from the combination and moved next to CNTN5.



Intersected Significant Snps

HTR3A_chr11:113842806

NRG1_chr8:31679387

Figure 7.12: The display surface designated into various areas.

or filtering a file with another object using the other hand. Markers were provided to help participants track gene expression files contained within unused objects (e.g. on the borders of the screen) (ATB T9).

As stated above, tools were provided for the participants to mark artefacts and to record the combinations explored and the resulting number of SNPs (ATB T10). The majority of participants (7 out of 10) were observed recording all possible combinations at the start of each task. The rest recorded their combinations after exploring them on the interface. The first approach was abandoned by the majority of participants in task 1 and then all participants in task 2 as the exploration progressed and groups and clusters were formed. A little under half of the participants spoke out loud to themselves when interacting with the objects (ATB B14). This was observed more frequently as the participants progressed to task 2, the more complicated task.

The types of comments received for the TUI and touch UI seems to shed some light on the users' concerns when interacting with both interfaces. While suggestions to improve and automate subtasks were given by the participants in both conditions, more participants from the tangible condition were inclined to share their ideas for improvements. Interestingly, their suggestions were closely related to the use of the tangible object. The response times when interacting with the TUI and touch UI were nearly identical, but only the participants in the touch condition made comments about response speed. This may be related to the physicality of the tangible objects; the objects inhabit the same space as the user and thus are thought to be more responsive compared to the touch condition.

7.5 Summary

In this chapter, a user experiment compared the performance of the TUI against a baseline touch UI. Users' interactions, explorations and preference were also logged and observed. The experiment's participants, materials and procedures were described (see Section 7.2). Objective and subjective metrics were examined using the interaction logs, exploration logs and observations (see Sections 7.3.1 and 7.3.2). Participants answers to the USE questionnaire and comments were also described (see Section 7.3.3). The results of the experiment showed that bimanual interactions were infrequently utilised. They also showed that participants found patterns faster with the TUI than with the touch UI as they adopted more effective strategies and performed fewer unnecessary analyses. This was because epistemic actions were adopted early in the task and maintained throughout the explorations.

Chapter 8

Conclusions and Future Work

8.1 Introduction

This thesis addresses the development of a TUI for infovis. Requirements are first elicited for the infovis case study using interviews and observations of the analysis scenario. The design options for a touch UI and a TUI are explored with the aid of well-established frameworks and classifications. To combine the strengths of both approaches, a system that balances touch and tangible interactions is envisioned. To determine technical feasibility, two common TUI technologies are systematically evaluated and tag-based computer vision is found to be most suitable for the infovis TUI. This has led to the iterative development of a TUI, as well as a baseline touch UI. A user experiment is conducted to compare the TUI with the touch UI. The results show that participants found patterns faster with the TUI than the touch UI, as they adopted more effective strategies and performed fewer unnecessary analyses. This chapter completes this thesis by describing conclusions and putting forward suggestions for future work.

8.2 Summaries

The overall aim of the research was to design, develop and evaluate a TUI for infovis, using eQTL analysis as a case study. Existing tangible systems for infovis are largely developed for smaller data sets (e.g. [40, 87, 111, 133, 147, 169, 204, 232]), limit the number of tangibles to a small set that are used as reconfigurable tools (e.g.

[87, 169, 211, 214] or limited to a small set of tasks (e.g. [53, 149, 191, 204, 227]). Also, little previous research has conducted user experiments to evaluate user performance, and even fewer reflected on the effect of TUI users' performance, particularly the effects of bimanual interaction and epistemic actions. The rest of this section describes this research's conclusions in the context of the research questions (see Section 1.2).

What is the analysis workflow for an interactive infovis case study?

For the purpose of developing a TUI for visualising eQTL, semi-structured interviews were conducted with four analysts to gain a broader understanding of the analysis process and elicit interface requirements. Five design implications were identified: scale, comparison, heterogeneity, sharing, and visualisation. An analysis scenario was also explored with the analysts to outline functional requirements for eQTL analysis, especially where data visualisation plays an important role. Nine functional requirements were outlined and formed the basis for a graphical application for eQTL visualisation (see Table 4.2).

One of the nine functionalities elicited, combining files to facilitate the detection of patterns from multiple eQTL result files, was itself novel. This requirement was developed using a typical grouping approach where two lists are maintained and files were swapped between them to form a combination from a whole. A user experiment was conducted to investigate the effect of input type (touch and mouse), font size (12 pt and 20 pt) and original list size (10, 15 and 20 items) on forming combinations of five items. The purpose of this study was to determine the strengths and weakness of touch interaction and the combination approach that needs to be taken into account when developing the baseline touch UI and the experimental task. The experiment's results showed that two factors affected the combination approach considered and touch input: size of the touch contact and the list size.

How should a TUI for interactive infovis be designed?

One of the main contributions of this work is the exploration of the design options for a touch UI and a TUI for infovis. The designs for both interfaces were considered within the context of well-established frameworks for the abstraction and design of touch and tangible interfaces.

The touch UI design options exploited desktop paradigms and abstract interactions due to their familiarity and users' preference for this form of interaction with touch interfaces [56, 264]. Gestures are common means for communication in real life settings and hold promise for touch input, including direct manipulation and space-distributed interaction. Interactive surface touch gestures were classified along four dimensions [264], of which nature and flow were used to categorise gestures for the touch UI. Handedness was also considered for various gestures to utilise the benefits of bimanual interactions when possible. The potential benefits and limitations of these various classifications were discussed to identify strengths and weaknesses of the various interactions within the context of a touch UI for infovis. Based on this discussion, the final touch UI combined desktop idioms with common gestures for the eQTL tasks.

An interactive surface TUI consisted of a tabletop and a number of tangible objects. The meaning of tangible objects lie along a continuum of object meanings [250], where at the centre of the continuum are objects that are the physical representation of their digital counterpart. Moving to the left or right of that centre, object's are either stripped of what can be done with them or become more generic and abstract. The TUI's objects were examined at various points on the continuum to determine potential benefits and limitations. The design options for the TUI were explored using the TAC paradigm, where objects and constraints are identified and behaviours determined based on their relationship [212]. An argument for and against object specificity was discussed, which led to initial designs for an exclusively tangible TUI.

Design options considered for the touch UI and TUI were then combined to form an initial hybrid TUI. The TUI aimed to utilise the two forms of interaction within the context of the eQTL tasks by balancing the strengths and limitations of touch and tangible interaction. The interactions adopted for the TUI helped identify three modalities to be sensed by the tangible objects: position, orientation and stack.

What technologies are suitable for infovis TUIs?

The next contribution, evaluated the three modalities using two common TUI development technologies: micro-controllers and tag-based computer vision [210].

Micro-controllers and sensors allowed for the development of stand-alone tangible objects (active tangibles) that did not require direct contact with an interactive surface to function, thereby expanding the interaction design space to above and around the surface (e.g. stacking objects to form combinations). A stand-alone tangible object utilising a micro-controller, accelerometer, gyroscope and force sensor was developed and systematically evaluated to determine the sensors' accuracy at detecting the three modalities. For single-axis displacements not exceeding 40 cm, the position of an object was accurately detected by an accelerometer with a margin of error less than 1 cm. However, dual-axes displacements exceeding 20 cm accumulated error that at times grew greater than 10%. For the detection of an object's orientation using an accelerometer and gyroscope, the sensors proved accurate at determining the face upon which a cube object was placed. Force sensors were only found to be reliable for the detection of one objects stacked on top of another.

Tag-based computer vision used fiducial markers on tangible objects to uniquely detect their position and orientation in real time, and is commonly used for infovis TUIs (e.g. [114, 124, 248, 253]). The interactive surface utilised for this research (the Samsung SUR40) is a rear-projected MBV system that detects objects on its surface using byte tags. A cube object was marked with a tag and systematically evaluated to determine the system's accuracy for the detection of position and orientation. The object's position was accurately detected rapidly with negligible margin of error (less than 1 cm). For the detection of the z-axis orientation, the tagged object also proved accurate (mean error was less than 1°). In order to detect force (i.e. stacking) computer vision would have required additional technologies (e.g. [16]) or customised markers (e.g. [15]) that restrict the object's shape and size.

Despite the benefits of utilising active tangibles, they proved inaccurate for the detection of an object's position. Computer vision was thus deemed the more appropriate technical approach for the TUI. The TUI was adjusted to eliminate stacking, which was not directly supported by the tag-based computer vision system. An initial TUI consisted of a Samsung SUR40, an eQTL visualisation application and tangible objects. The system was informally evaluated and the feedback received from the users was used to refine the final TUI. The final TUI for infovis was developed for the purpose of interactively engaging users in the interpretation of eQTL analysis results. A baseline touch UI was also developed based on the design options explored previously. The touch UI was used to comparatively asses the strengths and limitations of the TUI for infovis.

How effective is an infovis TUI compared with a touch UI baseline? The final contribution of this thesis is a user experiment that compared the TUI against the baseline touch UI in an eQTL exploration task. A between-participants design was adopted, and two hypotheses were formulated. First, the TUI would promote bimanual interaction more frequently than the touch UI and therefore reduce the time needed to explore gene expression results and combinations. Second, epistemic actions would be adopted more often in the TUI compared to the touch UI and would result in more efficient data exploration. Twenty participants were asked to explore significance for single gene expression files as well as combinations where interactions were logged and observed. The experiment concluded with a USE questionnaire which gathered subjective feedback about the usability of the systems, as well comments regarding the task and system usability.

The experiment showed that participants completed the tasks significantly faster with the TUI compared to the touch UI. The difference in performance was investigated in two separate analyses: time on sub-task and number of combinations that participants checked. Number of combinations checked by participant for the TUI was found to be fewer than those checked for the touch UI. Further examination identified three strategies for exploration: breadth-first, depth-first and mixed. The majority of the TUI participants adopted the more efficient breadth-first approach, while touch UI participants dividedly adopted all three strategies. The adoption of the more efficient strategy reduced the number of unnecessary explorations. The touch UI participants also suffered more inefficiencies, including: repetition and unnecessary exploration due to strategy or losing track of combinations explored.

Subjective preferences indicated that participants found the TUI to be slightly more useful, easier to use, easier to learn and satisfactory, while the touch UI was rated higher for the time it took to learn and simplicity. The participants' comments found the TUI user friendly and easy to use, and other participants voiced their preference for the TUI over touchscreens in general. Participants also had concerns regarding the system, particularly the repetitiveness of the task and suggested the utilisation of a new object for the TUI or menu items for the touch UI that would aid users in performing tasks that are repetitive in nature.

Bimanual interaction is thought to be promoted by natural interactions, such as touch and tangible. However, few participants took advantage of the bimanual interactions for the touch UI and TUI, which is consistent with previous findings [234, 235, 240]. When bimanual interactions were utilised they were largely asymmetric or concurrent unimanual. Epistemic actions were strongly utilised by the participants of the TUI condition compared to the touch UI. Findings from previous studies suggested the promotion of epistemic actions by TUIs for problem solving tasks [7, 8], but these investigations were rarely undertaken in TUIs for infovis research. In the experiment, the majority of TUI participants manipulated and spatially arranged tangible objects around the interactive surface. Objects were regularly grouped according to significance, while also being placed within user-devised divisions on the interactive surface.

8.3 Future Work

The are several ways in which this research can be continued. In the short term, modifications to the current version of the TUI will possibly enhance performance and in particular those voiced by the second user experiment's participants (see Section 7.3.3). In the longer run, the present research lends itself to several directions for future work.

8.3.1 Expanding on eQTL Tasks

The eQTL tasks elicited from quantitative genetics analysts were based on the interviews and the analysis scenario (see Section 4.2). While the majority were addressed by the developed systems, the capability to share analysis results in public or private databases was not implemented from early on. This was to focus on the part of eQTL analysis where data visualisation plays an important role. Tangible objects in an infovis TUI lend themselves to being shared between analysts in different labs in the same facility by sharing the physical object. Other infovis TUIs have devised means to store or record analysis results (e.g.

[55, 147, 191, 227, 250]), however records were not shared with other collaborators and were retrieved on the same system.

The eQTL filtering task detects genetic variants that meet a particular significance threshold. These genetic variants are typically functional, i.e. directly link a person's disease susceptibility or drug response to a specific allele. This, of course, means that the carriers of that allele in the genetic variant have an increased risk of developing the disease, while other alleles mean a lessened risk. Nevertheless, many genetic variants are not functional but are of close proximity and associated with a functional variant. Neighbouring alleles that are found together more often than expected under statistical independence are said to be in linkage disequilibrium (LD) [222]. The patterns of SNP alleles that are inherited together are known as haplotypes. The international HapMap project [72] has developed a map of the human haplotype from four large populations of African, Asian and European descents, which substantially covers the genetic variations found in the world's populations. The results of an eQTL analysis can be combined with hapmap project results. When a user filters the combined results for significant SNPs, the results can include variants that meet the significance threshold as well as potentially functional variants that are in LD.

8.3.2 Expanding on eQTL Visualisation

The results of an eQTL study are typically shown in a Manhattan plot or summary table. Other visualisations and tools, e.g eQTL Viewer [272] and eQTL Explorer [168], were presented to the analysts interviewed in Section 4.2. One of the analysts explained that these tools may be useful for biologist but are not particularly well suited for analysts. The analysts continued to acknowledge the importance of plotting (using the Manhattan plots) and that finding a way to combine and contrast plots could simplify the analysis process. This suggestion was addressed with the infovis tools that were developed. However it poses the question if whether other ways of presenting visualisation should also be explored. This exploration will particularly be useful when working directly with analysts, where various presentations are examined for use and exploited for all that they may offer to the task.

8.3.3 Enhancing TUI

The final infovis TUI consisted of an eQTL visualisation application, an interactive surface (Samsung SUR40) and two types of tangible objects. The technology used was computer vision, since micro-controllers did not prove suitable for the design that was envisioned (see Section 6.2). Computer vision is popularly used for infovis TUIs (e.g. [111, 169, 211, 253]), as they tend to adopt interactive displays to present infovis.

During the experiment comparing TUI and touch UI (see Chapter 7), several limitations were observed during interaction with the computer vision system, which were not voiced by the participants of either condition. Detection problems were noted at times, despite adhering to the tag's printing guidelines provided by Microsoft [166] and shielding detection from contact made on top of the object being handled (see Section 6.5.1). In one instance, the object was placed on the surface but its digital representation was inconsistent, i.e. flickering. To stop this, the user picked up the object and placed it on the surface again where it continued to display the digital representation consistently. To improve system performance and limit the occurrence of these types of problem, another experiment can be carried out to identify these problems and explore the best quality tag to use. Alternatively, a Kalman filter [117] can be used to further improve the accuracy of detection. The filter can also be used to estimate and reassign the position and orientation of a byte tag that lost contact with the tabletop.

Other infovis TUIs utilise micro-controllers, sensors and actuators (e.g. [9, 40, 245, 253]). Sifteo cubes [164] are commercial active tangibles that combine a tangible cube with a display and a graphical interface. The cubes can detect tapping, tilting, shaking, flipping and neighbouring. They are used with one of the infovis TUI's for casual health information [40]. Integrating Sifteo cubes into the current infovis TUI for eQTL holds the potential of improving interaction, e.g. combine files with the more intuitive stacking instead of tapping. However, there are still some issues to take into account due to the rigid structure of the cubes. Other research [9] used smart watches to allow for touch interaction with the tangibles. This research used passive tangibles due the results of the evaluations that found some sensors used with micro-controllers inferior to computer visions (see Section 6.2), but adopting commercialised products can help extend the design space and achieve more intuitive interactions. Relatively little research addresses the difference between passive and active tangibles, so this can also offer an opportunity to assesses the active TUI against the current passive TUI.

Participants in the TUI condition had several comments and suggestions (see Section 7.3.3) that can help improve the usability of the tool. To form a combination of gene expression files, one of the files was used as a base where other files could be combined to. This approach was taken due to the decision of using only gene expression objects instead of including another object type. However, a participant explained that the use of the base object hindered his exploration during the experiment. Other participants suggested that the threshold value of a SNP object can be saved so that repetitive tasks are easier to complete (e.g. using the same filtering threshold to collect significant SNPs across multiple files, instead of rotating the dial each time). This is possible by providing an option to have the threshold value be set once and reused.

8.3.4 Overcoming Experiment's Limitations

There are several limitations with the experiment carried out in Chapter 7 that compared the final infovis TUI against a touch UI version. Participants recruited for the experiment were from the general public, with the majority being students (see Section 7.2). The experiment did not take into account users' preconceptions. It was assumed that the diversity of their pre-existing knowledge would be randomly distributed between both conditions. Future experiments should have better control over this dimension of the experiment. Also all participants, with one exception, had little background in quantitative genetics and future studies can benefit from getting feedback from eQTL analysts. Participants' epistemic actions were collected via observation and screenshots of user's interactions. While these did suffice to capture differences between touch and tangible interaction for the experimental task, future experiments should benefit from video recording that can better be analysed using the ATB framework [57]).

8.4 Conclusion

The overall aim of this research was to design, develop and evaluate a TUI for infovis, with eQTL analysis as the visualisation case study. In this work, functional requirements for a TUI for eQTL infovis were extracted from semi-structured interviews with experts. One of the main contributions of this thesis was the exploration of design options for touch and tangible infovis interfaces within the context of well-established frameworks. These options were then combined to form a hybrid TUI that balanced touch and tangible interactions. The following contribution evaluated two common TUI implementation technologies, microcontroller sensors and tag-based computer vision, to determine the most suited technology for the designed hybrid TUI. The final contribution compared the TUI against a baseline touch UI in a user experiment, where the results showed that participants adopted more effective strategies and performed fewer unnecessary analyses with the TUI compared to the touch UI.

The findings of this research's investigation in developing a TUI for infovis can have implications for other researchers considering developing a TUI for infovis that utilises an interactive surface and tangible objects. Similar approaches could be performed to explore design space, technical feasibility and development. The promising outcome of the comparative experiment showed that this research should encourage the examination of bimanual interaction and epistemic actions for various tasks to contextually identify performance benefits.

Opportunities for future work include short and long term enhancements to the design and development of the TUI infovis, as well as occasions for further evaluation. Active tangible (e.g. Sifteo cubes [164]) can be integrated to the current implementation to potentially enhance interactions. A comparative assessment of active and passive tangibles can shed lights on their use within the context of infovis TUIs. Field experimentation with eQTL experts (e.g. utilising field logs and observations) can elicit usage statistics unique to domain experts, as well as identify behaviours for further study and improvement. This can in turn expand the implementation to include eQTL tasks not considered in the current implementation.

References

[1] Agarawala, A., and Balakrishnan, R. Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2006), CHI '06, ACM, pp. 1283–1292.

[2] Aguilar-Garcia, A., Fortes, S., Colin, E., and Barco, R. Enhancing RFID indoor localization with cellular technologies. *EURASIP Journal on Wireless Communications and Networking 2015*, 1 (2015), 1–22.

[3] Alarcon, M., Abrahams, B. S., Stone, J. L., Duvall, J. A., Perederiy, J. V., Bomar, J. M., Sebat, J., Wigler, M., Martin, C. L., Ledbetter, D. H., Nelson, S. F., Cantor, R. M., and Geschwind, D. H. Linkage, association, and gene-expression analyses identify CNTNAP2 as an autism-susceptibility gene. *The American Journal of Human Genetics 82*, 1 (2008), 150–159.

[4] Alibali, M. W., and DiRusso, A. A. The function of gesture in learning to count: more than keeping track. *Cognitive Development 14*, 1 (1999), 37 – 56.

[5] Amar, R., Eagan, J., and Stasko, J. Low-level components of analytic activity in information visualization. In *Proceedings of the IEEE Symposium on Information Visualization* (2005), INFOVIS '05, pp. 111–117.

[6] Anderson, D., Frankel, J. L., Marks, J., Leigh, D., Sullivan, E., Yedidia, J., and Ryall, K. Building virtual structures with physical blocks. In *Proceedings of the 12th Annual ACM Symposium on User Interface*

Software and Technology (New York, NY, USA, 1999), UIST '99, ACM, pp. 71–72.

[7] ANTLE, A. N. Exploring how children use their hands to think: An embodied interactional analysis. *Behaviour and Information Technology 32*, 9 (2013), 938–954.

[8] ANTLE, A. N., AND WANG, S. Comparing motor-cognitive strategies for spatial problem solving with tangible and multi-touch interfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2013), TEI '13, ACM, pp. 65–72.

[9] ARIF, A. S., MANSHAEI, R., DELONG, S., EAST, B., KYAN, M., AND MAZALEK, A. Sparse tangibles: Collaborative exploration of gene networks using active tangibles and interactive tabletops. In *Proceedings of the 10th International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2016), TEI '16, ACM, pp. 287–295.

[10] BAILLY, G., NIGAY, L., AND AUBER, D. NAVRNA: Visualization - exploration - editing of RNA. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2006), AVI '06, ACM, pp. 504–507.

[11] BALAKRISHNAN, R., AND HINCKLEY, K. Symmetric bimanual interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2000), CHI '00, ACM, pp. 33–40.

[12] BALLAGAS, R., RINGEL, M., STONE, M., AND BORCHERS, J. istuff: A physical user interface toolkit for ubiquitous computing environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2003), CHI '03, ACM, pp. 537–544.

[13] BANISSI, E., MARCHESE, F. T., AND FORSELL, C. *Information Visualisation: Techniques, Usability and Evaluation.* Cambridge Scholars Publishing, 2014.

[14] Barrett, J. C. Haploview: Visualization and analysis of SNP genotype data. *Cold Spring Harbor Protocols 2009*, 10 (2009), pdb.ip71.

[15] Bartindale, T., and Harrison, C. Stacks on the surface: Resolving physical order using fiducial markers with structured transparency. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2009), ITS '09, ACM, pp. 57–60.

[16] Baudisch, P., Becker, T., and Rudeck, F. Lumino: Tangible blocks for tabletop computers based on glass fiber bundles. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, ACM, pp. 1165–1174.

[17] Baur, D., Lee, B., and Carpendale, S. Touchwave: Kinetic multitouch manipulation for hierarchical stacked graphs. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2012), ITS '12, ACM, pp. 255–264.

[18] Bdeir, A. Electronics as material: Littlebits. In *Proceedings of the 3rd International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2009), TEI '09, ACM, pp. 397–400.

[19] Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. Toolglass and magic lenses: The see-through interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1993), SIGGRAPH '93, ACM, pp. 73–80.

[20] Billinghurst, M., Cheok, A., Prince, S., and Kato, H. Real world teleconferencing. *IEEE Computer Graphics and Applications 22*, 6 (2002), 11–13.

[21] Blum, J. *Exploring Arduino.* Wiley, 2013.

[22] Brehmer, M., and Munzner, T. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2376–2385.

[23] Brooker, G., and Brooker, A. T. *Introduction to sensors for ranging and imaging.* SciTech Pub. Incorporated, 2009.

[24] Bruder, C. E., Piotrowski, A., Gijsbers, A. A., Andersson, R., Erickson, S., de StÃhl, T. D., Menzel, U., Sandgren, J., von Tell, D., Poplawski, A., Crowley, M., Crasto, C., Partridge, E. C., Tiwari, H., Allison, D. B., Komorowski, J., van Ommen, G.-J. B., Boomsma, D. I., Pedersen, N. L., den Dunnen, J. T., Wirdefeldt, K., and Dumanski, J. P. Phenotypically concordant and discordant monozygotic twins display different DNA copy-number-variation profiles. *The American Journal of Human Genetics 82*, 3 (2008), 763–771.

[25] Buxton, W. Combined keyboard/touch tablet input device. *XEROX Disclosure Journal 19*, 2 (1994), 109–111.

[26] Buxton, W. Multi-touch systems that i have known and loved. `http://www.billbuxton.com/multitouchOverview.html`, 2007 (accessed Decemeber 22, 2014).

[27] Buxton, W. 31.1: Invited paper: A touching story: A personal perspective on the history of touch interfaces past and future. In *Sid symposium digest of technical papers* (2010), vol. 41, Wiley Online Library, pp. 444–448.

[28] Buxton, W., and Myers, B. A study in two-handed input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1986), CHI '86, ACM, pp. 321–326.

[29] Cairns, P., and Cox, A. L. *Research Methods for Human-Computer Interaction.* Cambridge University Press, 2008.

[30] Camarata, K., Do, E. Y.-L., Johnson, B. R., and Gross, M. D. Navigational blocks: Navigating information space with tangible media. In *Proceedings of the 7th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2002), IUI '02, ACM, pp. 31–38.

[31] Card, S. K., English, W. K., and Burr, B. J. Human-computer interaction. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987, ch. Evaluation of Mouse, Rate-controlled Isometric Joystick, Step Keys, and Text Keys, for Text Selection on a CRT, pp. 386–392.

[32] Card, S. K., Mackinlay, J. D., and Shneiderman, B. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[33] Casalta, D., Guiard, Y., and Beaudouin-Lafon, M. Evaluating two-handed input techniques: Rectangle editing and navigation. In *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 1999), CHI EA '99, ACM, pp. 236–237.

[34] Chan, L., Müller, S., Roudaut, A., and Baudisch, P. Capstones and zebra widgets: Sensing stacks of building blocks, dials and sliders on capacitive touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), CHI '12, ACM, pp. 2189–2192.

[35] Chen, X., McClelland, M., Jia, Z., Rahmatpanah, F. B., Sawyers, A., Trent, J., Duggan, D., and Mercola, D. The identification of trans-associations between prostate cancer GWAS SNPs and RNA expression differences in tumor-adjacent stroma. *Oncotarget 6*, 3 (2015), 1865.

[36] Chen, Y., Zhu, J., Lum, P. Y., Yang, X., Pinto, S., MacNeil, D. J., Zhang, C., Lamb, J., Edwards, S., Sieberts, S. K., et al. Variations in DNA elucidate molecular networks that cause disease. *Nature 452*, 7186 (2008), 429–435.

[37] Chin, J. P., Diehl, V. A., and Norman, K. L. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1988), CHI '88, ACM, pp. 213–218.

[38] Chuah, M. C., and Roth, S. F. On the semantics of interactive visualizations. In *Proceedings of the IEEE Symposium on Information Visualization* (Washington, DC, USA, 1996), INFOVIS '96, IEEE Computer Society, pp. 29–36.

[39] Cirelli, M., and Nakamura, R. A survey on multi-touch gesture recognition and multi-touch frameworks. In *Proceedings of the ACM International*

Conference on Interactive Tabletops and Surfaces (New York, NY, USA, 2014), ITS '14, ACM, pp. 35–44.

[40] Claes, S., Coenen, J., Slegers, K., and Vande Moere, A. Design implications of casual health visualization on tangible displays. In *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2015), CHI EA '15, ACM, pp. 1839–1844.

[41] Cockburn, A., Ahlström, D., and Gutwin, C. Understanding performance in touch selections: Tap, drag and radial pointing drag with finger, stylus and mouse. *International Journal of Human-Computer Studies 70*, 3 (2012), 218 – 233.

[42] Dao, T. H., Nguyen, Q. C., Ngo, V. D., Le, M. T., and Hoang, C. A. Indoor localization system based on passive RFID tags. In *Proceedings of the 5th International Conference on Intelligent Systems, Modelling and Simulation* (2014), ISMS '14, pp. 579–584.

[43] De Cruz, H. How do spatial representations enhance cognitive numerical processing? *Cognitive Processing 13*, 1 (2012), 137–140.

[44] Destefano, M., Lindstedt, J. K., and Gray, W. D. Use of complementary actions decreases with expertise. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society* (2011), pp. 2709–2014.

[45] Dicati, R. The ancients' astronomy. In *Stamping Through Astronomy*. Springer, 2013, pp. 19–55.

[46] Dietz, P., and Leigh, D. Diamondtouch: A multi-user touch technology. In *Proceedings of the 14th annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2001), UIST '01, ACM, pp. 219–226.

[47] Dillon, R. F., Edey, J. D., and Tombaugh, J. W. Measuring the true cost of command selection: Techniques and results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1990), CHI '90, ACM, pp. 19–26.

[48] Dix, A. *Human-Computer Interaction.* Springer, 2009.

[49] Dix, A., and Ellis, G. Starting simple: Adding value to static visualisation through simple interaction. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 1998), AVI '98, ACM, pp. 124–134.

[50] Djajadiningrat, T., Wensveen, S., Frens, J., and Overbeeke, K. Tangible products: Redressing the balance between appearance and action. *Personal Ubiquitous Computing 8*, 5 (2004), 294–309.

[51] Don, L., and Smith, S. P. Applying bimanual interaction principles to text input on multi-touch surfaces and tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2010), ITS '10, ACM, pp. 253–254.

[52] Du, J., Gerdtman, C., and Lindén, M. Signal processing algorithms for position measurement with mems-based accelerometer. In *Proceedings of the 16th Nordic-Baltic Conference on Biomedical Engineering* (2015), Springer, pp. 36–39.

[53] Dumas, B., Moerman, B., Trullemans, S., and Signer, B. ArtVis: Combining advanced visualisation and tangible interaction for the exploration, analysis and browsing of digital artwork collections. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2014), AVI '14, ACM, pp. 65–72.

[54] Duschl, R. Science education in three-part harmony: Balancing conceptual, epistemic, and social learning goals. *Review of Research in Education 32*, 1 (2008), 268–291.

[55] Ebert, A., Weber, C., Cernea, D., and Petsch, S. TangibleRings: Nestable circular tangibles. In *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI EA '13, ACM, pp. 1617–1622.

[56] Epps, J., Lichman, S., and Wu, M. A study of hand shape use in tabletop gesture interaction. In *Proceedings of the ACM Conference Extended*

Abstracts on Human Factors in Computing Systems (New York, NY, USA, 2006), CHI EA '06, ACM, pp. 748–753.

[57] Esteves, A., Bakker, S., Antle, A. N., May, A., Warren, J., and Oakley, I. The ATB framework: Quantifying and classifying epistemic strategies in tangible problem-solving tasks. In *Proceedings of the 9th International Conference on Tangible, Embedded, and Embodied Interaction* (New York, NY, USA, 2015), TEI '15, ACM, pp. 13–20.

[58] Esteves, A., van den Hoven, E., and Oakley, I. Physical games or digital games?: Comparing support for mental projection in tangible and virtual representations of a problem-solving task. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2013), TEI '13, ACM, pp. 167–174.

[59] Ferreira, J., Barr, P., and Noble, J. The semiotics of user interface redesign. In *Proceedings of the 6th Australasian Conference on User Interface* (Darlinghurst, Australia, 2005), AUIC '05, Australian Computer Society, Inc., pp. 47–53.

[60] Findlater, L., Froehlich, J. E., Fattal, K., Wobbrock, J. O., and Dastyar, T. Age-related differences in performance with touchscreens compared to traditional mouse input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI '13, ACM, pp. 343–346.

[61] Fishkin, K. P. A taxonomy for and analysis of tangible interfaces. *Personal Ubiquitous Computing 8*, 5 (2004), 347–358.

[62] Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology 47*, 6 (1954), 381.

[63] Fitzmaurice, G. W. *Graspable user interfaces.* PhD thesis, Citeseer, 1996.

[64] Fitzmaurice, G. W., Ishii, H., and Buxton, W. Bricks: Laying the foundations for graspable user interfaces. In *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems (New York, NY, USA, 1995), CHI '95, ACM, pp. 442–449.

[65] Fludd, R. Great chain of being. `http://farm7.static.flickr.com/6187/6130074265_d9eee51380_o.jpg`, 2014 (accessed December 13, 2015).

[66] Follmer, S., Leithinger, D., Olwal, A., Hogge, A., and Ishii, H. inform: Dynamic physical affordances and constraints through shape and object actuation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2013), UIST '13, ACM, pp. 417–426.

[67] Forlines, C., Wigdor, D., Shen, C., and Balakrishnan, R. Direct-touch vs. mouse input for tabletop displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), CHI '07, ACM, pp. 647–656.

[68] Forsyth, D. A., and Ponce, J. *Computer Vision: A Moden Approach.* Pearson, 2011.

[69] Franke, L., and Jansen, R. C. *Cardiovascular Genomics*, vol. 573 of *Methods in Molecular Biology.* Humana Press, 2009, pp. 311–328.

[70] Fujii, K., Shimamura, J., Arakawa, K., and Arikawa, T. Tangible search for stacked objects. In *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2003), CHI EA '03, ACM, pp. 848–849.

[71] Gaukrodger, S. J., and Lintott, A. Augmented reality and applications for assistive technology. In *Proceedings of the 1st International Convention on Rehabilitation Engineering and Assistive Technology: In Conjunction with 1st Tan Tock Seng Hospital Neurorehabilitation Meeting* (New York, NY, USA, 2007), i-CREATe '07, ACM, pp. 47–51.

[72] Gibbs, R. A., Belmont, J. W., Hardenbol, P., Willis, T. D., Yu, F., Yang, H., Ch'ang, L.-Y., Huang, W., Liu, B., Shen, Y., et al. The international HapMap project. *Nature 426*, 6968 (2003), 789–796.

[73] Gibson, J. J. *The Ecological Approach to Visual Perception: Classic Edition.* Psychology Press, 2014.

[74] Gilbert, D. The JFreeChart class library. *Developer Guide. Object Refinery 7* (2002).

[75] Gillet, A., Sanner, M., Stoffler, D., and Olson, A. Tangible interfaces for structural molecular biology. *Structure 13*, 3 (2005), 483–491.

[76] Glover, B., and Bhatt, H. *RFID essentials.* O'Reilly, 2006.

[77] Goldin-Meadow, S. Talking and thinking with our hands. *Current Directions in Psychological Science 15*, 1 (2006), 34–39.

[78] Graham, E. D., and MacKenzie, C. L. Physical versus virtual pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1996), CHI '96, ACM, pp. 292–299.

[79] Greenberg, S., and Boyle, M. Customizable physical interfaces for interacting with conventional applications. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2002), UIST '02, ACM, pp. 31–40.

[80] Greenberg, S., and Fitchett, C. Phidgets: Easy development of physical interfaces through physical widgets. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2001), UIST '01, ACM, pp. 209–218.

[81] Guiard, Y. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behaviour 19* (1987), 486–517.

[82] Guthrie, L. G., and Vallée-Tourangeau, F. Interactivity and mental arithmetic: Coupling mind and world transforms and enhances performance. *Studies in Logic, Grammar and Rhetoric 41*, 1 (2015), 41–59.

[83] Gutierrez, M., Vexo, F., and Thalmann, D. *Stepping into virtual reality.* Springer Science & Business Media, 2008.

[84] Hagen, H., Ebert, A., van Lengen, R. H., and Scheuermann, G. Scientific visualization: Methods and applications. In *Proceedings of the 19th Spring Conference on Computer Graphics* (New York, NY, USA, 2003), SCCG '03, ACM, pp. 23–33.

[85] Hahnel, D., Burgard, W., Fox, D., Fishkin, K., and Philipose, M. Mapping and localization with rfid technology. In *Proceedings of the IEEE International Conference on Robotics and Automation* (2004), vol. 1 of *ICRA '04*, pp. 1015–1020 Vol.1.

[86] Hallnäs, L., and Redström, J. From use to presence: On the expressions and aesthetics of everyday computational things. *ACM Transactions on Computer-Human Interaction 9*, 2 (2002), 106–124.

[87] Hancock, M., Hilliges, O., Collins, C., Baur, D., and Carpendale, S. Exploring tangible and direct touch interfaces for manipulating 2D and 3D information on a digital table. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2009), ITS '09, ACM, pp. 77–84.

[88] Hancock, M. S., Carpendale, S., Vernier, F. D., Wigdor, D., and Shen, C. Rotation and translation mechanisms for tabletop interaction. In *Proceedings of the 1st IEEE International Workshop on Horizontal Interactive Human-Computer Systems* (Washington, DC, USA, 2006), TABLETOP '06, IEEE Computer Society, pp. 79–88.

[89] Harrell, J. A., and Brown, V. M. The world's oldest surviving geological map: The 1150 BC turin papyrus from Egypt. *The Journal of Geology* (1992), 3–18.

[90] Hartmann, B., Klemmer, S. R., Bernstein, M., and Mehta, N. d.tools: Visually prototyping physical UIs through statecharts. In *Proceedings of the 19th annual ACM symposium on User Interface Software and Technology* (2005), UIST '05, ACM.

[91] Hearst, M. A. Information visualization: Principles, promise, and pragmatics. `http://people.ischool.berkeley.edu/~hearst/research/infovis.html`, 2003 (accessed September 26, 2014).

[92] Heer, J., and Shneiderman, B. Interactive dynamics for visual analysis. *Communications of the ACM 55*, 4 (2012), 45–54.

[93] Heijboer, M., and van den Hoven, E. Keeping up appearances: Interpretation of tangible artifact design. In *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges* (New York, NY, USA, 2008), NordiCHI '08, ACM, pp. 162–171.

[94] Hekimian-Williams, C., Grant, B., Liu, X., Zhang, Z., and Kumar, P. Accurate localization of RFID tags using phase difference. In *Proceedings of The IEEE International Conference on RFID* (2010), pp. 89–96.

[95] Hermann, M., and Weber, M. When three worlds collide: A model of the tangible interaction process. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7* (New York, NY, USA, 2009), OZCHI '09, ACM, pp. 341–344.

[96] Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. Passive real-world interface props for neurosurgical visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1994), CHI '94, ACM, pp. 452–458.

[97] Hinrichs, U., Carpendale, S., Scott, S. D., and Pattison, E. Interface currents: Supporting fluent collaboration on tabletop displays. In *Proceedings of the 5th International Conference on Smart Graphics* (Berlin, Heidelberg, 2005), SG '05, Springer-Verlag, pp. 185–197.

[98] Holmquist, L. E., Redström, J., and Ljungstrand, P. Token-based access to digital information. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing* (London, UK, UK, 1999), HUC '99, Springer-Verlag, pp. 234–245.

[99] Hooten, E. R., and Adams, J. Comparing input error for mouse and touch input. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (2011), SMC '11, pp. 2853–2858.

[100] Hornecker, E., and Buur, J. Getting a grip on tangible interaction: A framework on physical space and social interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2006), CHI '06, ACM, pp. 437–446.

[101] Huang, Y., and Eisenberg, M. Easigami: Virtual creation by physical folding. In *Proceedings of the 6th International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2012), TEI '12, ACM, pp. 41–48.

[102] Hutchins, E. L., Hollan, J. D., and Norman, D. A. Direct manipulation interfaces. *Human Computer Interaction 1*, 4 (1985), 311–338.

[103] IEEE. Second IEEE Symposium on Biological Data Visualization. `http://biovis.net/year/2012/about.html`, 2012 (accessed May 11, 2012).

[104] iOS developer library. iOS human interface guidelines. `https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/MobileHIG.pdf`, 2014 (accessed March 2, 2016).

[105] Ishii, H., and Ullmer, B. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1997), CHI '97, ACM, pp. 234–241.

[106] Ishii, H., Wisneski, C., Brave, S., Dahley, A., Gorbet, M., Ullmer, B., and Yarin, P. ambientROOM: Integrating ambient media with architectural space. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1998), CHI '98, ACM, pp. 173–174.

[107] Jackson, B., Lau, T. Y., Schroeder, D., Toussaint, K., and Keefe, D. A lightweight tangible 3D interface for interactive visualization of thin fiber structures. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2802–2809.

[108] Jacob, R. J. K., Ishii, H., Pangaro, G., and Patten, J. A tangible interface for organizing information using a grid. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2002), CHI '02, ACM, pp. 339–346.

[109] Jansen, Y. *Physical and tangible information visualization.* PhD thesis, Université Paris Sud - Paris XI, 2014.

[110] Jansen, Y., Dragicevic, P., and Fekete, J.-D. Evaluating the efficiency of physical visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI '13, ACM, pp. 2593–2602.

[111] Jetter, H.-C., Gerken, J., Zöllner, M., Reiterer, H., and Milic-Frayling, N. Materializing the query with facet-streams: A hybrid surface for collaborative search on tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, ACM, pp. 3013–3022.

[112] Jin, G.-y., Lu, X.-y., and Park, M.-S. An indoor localization mechanism using active RFID tag. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing* (June 2006), vol. 1 of *SUTC '06*, pp. 40–43.

[113] Joho, D., Plagemann, C., and Burgard, W. Modeling RFID signal strength and tag detection for localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation* (2009), ICRA '2009, IEEE Computer Society, pp. 3160–3165.

[114] Jordà, S., Geiger, G., Alonso, M., and Kaltenbrunner, M. The reacTable: Exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2007), TEI '07, ACM, pp. 139–146.

[115] Jung, H., and Stolterman, E. Material probe: Exploring materiality of digital artifacts. In *Proceedings of the 5th International Conference on*

Tangible, Embedded, and Embodied Interaction (New York, NY, USA, 2011), TEI '11, ACM, pp. 153–156.

[116] Kabbash, P., Buxton, W., and Sellen, A. Two-handed input in a compound task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1994), CHI '94, ACM, pp. 417–423.

[117] Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering 82*, 1 (1960), 35–45.

[118] Kaltenbrunner, M., and Bencina, R. reacTIVision: A computer-vision framework for table-based tangible interaction. In *Proceedings of the 1st International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2007), TEI '07, ACM, pp. 69–74.

[119] Karam, M., and schraefel, M. C. A taxonomy of gestures in human computer interactions. Technical Report ECSTR-IAM05-009, University of Southampton, 2005.

[120] Karat, J., McDonald, J. E., and Anderson, M. A comparison of menu selection techniques: Touch panel, mouse and keyboard. *International Journal of Man-Machine Studies 25*, 1 (1986), 73–88.

[121] Kasday, L. R. Touch position sensitive surface, 1984. US Patent 4,484,179.

[122] Kastens, K. A., Liben, L. S., and Agrawal, S. Epistemic actions in science education. In *Spatial Cognition VI. Learning, Reasoning, and Talking about Space*. Springer, 2008, pp. 202–215.

[123] Kato, H., and Billinghurst, M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality* (1999), IWAR '99, pp. 85–94.

[124] Kim, K., and Elmqvist, N. Embodied lenses for collaborative visual queries on tabletop displays. *Information Visualization 11*, 4 (Oct. 2012), 319–338.

[125] Kim, M. J., and Maher, M. L. The impact of tangible user interfaces on spatial cognition during collaborative design. *Design Studies 29*, 3 (2008), 222 – 253.

[126] Kirk, D., Sellen, A., Taylor, S., Villar, N., and Izadi, S. Putting the physical into the digital: Issues in designing hybrid interactive surfaces. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology* (Swinton, UK, 2009), BCS-HCI '09, British Computer Society, pp. 35–44.

[127] Kirsh, D. Complementary strategies: Why we use our hands when we think. In *Proceedings of the 17th Annual Conference of the Cognitive Science Society* (1995), Centre for Philosophical Research, pp. 161–175.

[128] Kirsh, D. The intelligent use of space. *Artificial Intelligence 73*, 1–2 (1995), 31 – 68. Computational Research on Interaction and Agency, Part 2.

[129] Kirsh, D. *Situating Instructions.* New Bulgarian University Press, 2011.

[130] Kirsh, D., and Maglio, P. On distinguishing epistemic from pragmatic action. *Cognitive Science 18*, 4 (1994), 513–549.

[131] Klassner, F., and Anderson, S. D. Lego mindstorms: Not just for k-12 anymore. *IEEE Robotics & Automation Magazine 10*, 2 (2003), 12–18.

[132] Klemmer, S. R., Li, J., Lin, J., and Landay, J. A. Papier-mache: Toolkit support for tangible input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2004), CHI '04, ACM, pp. 399–406.

[133] Klum, S., Isenberg, P., Langner, R., Fekete, J.-D., and Dachselt, R. Stackables: Combining tangibles for faceted browsing. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2012), AVI '12, ACM, pp. 241–248.

[134] Koike, H., Nishikawa, W., and Fukuchi, K. Transparent 2-D markers on an LCD tabletop system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI '09, ACM, pp. 163–172.

[135] Konchada, V., Jackson, B., Le, T., Borazjani, I., Sotiropoulos, F., and Keefe, D. F. Supporting internal visualization of biomedical datasets via 3D rapid prototypes and sketch-based gestures. In *Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2011), I3D '11, ACM, pp. 214–214.

[136] Kosara, R., Hauser, H., and Gresh, D. L. An interaction view on information visualization. In *Proceedings of the European Association for Computer Graphics* (2003), EUROGRAPHICS '03, pp. 123–137.

[137] Kruszynski, K. J., and van Liere, R. Tangible props for scientific visualization: Concept, requirements, application. *Virtual Reality 13*, 4 (2009), 235–244.

[138] Kuniavsky, M. *Observing the User Experience: A Practitioner's Guide to User Research (Morgan Kaufmann Series in Interactive Technologies).* Morgan Kaufmann, San Francisco, CA, USA, 2003.

[139] Lam, H., Bertini, E., Isenberg, P., Plaisant, C., and Carpendale, S. Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics 18*, 9 (Sept 2012), 1520–1536.

[140] Lazar, J., Feng, J. H., and Hochheiser, H. *Research Methods in Human-Computer Interaction.* Wiley Publishing, 2010.

[141] Lee, J. C., Avrahami, D., Hudson, S. E., Forlizzi, J., Dietz, P. H., and Leigh, D. The calder toolkit: Wired and wireless components for rapidly prototyping interactive devices. In *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques* (New York, NY, USA, 2004), DIS '04, ACM, pp. 167–175.

[142] Leganchuk, A., Zhai, S., and Buxton, W. Manual and cognitive benefits of two-handed input: An experimental study. *ACM Transactions on Computer-Human Interaction 5*, 4 (1998), 326–359.

[143] Lewis, J. R. Sample sizes for usability tests: Mostly math, not magic. *Interactions 13*, 6 (2006), 29–33.

[144] Liang, L., Morar, N., Dixon, A. L., Lathrop, G. M., Abecasis, G. R., Moffatt, M. F., and Cookson, W. O. A cross-platform analysis of 14,177 expression quantitative trait loci derived from lymphoblastoid cell lines. *Genome Research 23*, 4 (2013), 716–726.

[145] Liu, Z., and Stasko, J. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 999–1008.

[146] Lu, F., Chen, X., and Ye, T. T. Performance analysis of stacked rfid tags. In *Proceedings of the IEEE International Conference on RFID* (2009), IEEE Computer Society, pp. 330–337.

[147] Luderschmidt, J., Dörner, R., Seyer, M., Friess, F., and Heimann, R. Vispol: An interactive tabletop graph visualization for the police. In *Mensch & Computer* (2012), pp. 63–72.

[148] Lund, A. M. Measuring usability with the use questionnaire. *STC Usability SIG Newsletter 8*, 2 (2001).

[149] Ma, J., Sindorf, L., Liao, I., and Frazier, J. Using a tangible versus a multi-touch graphical user interface to support data exploration at a museum exhibit. In *Proceedings of the 9th International Conference on Tangible, Embedded, and Embodied Interaction* (New York, NY, USA, 2015), TEI '15, ACM, pp. 33–40.

[150] Macaranas, A., Antle, A. N., and Riecke, B. E. Bridging the gap: Attribute and spatial metaphors for tangible interface design. In *Proceedings of the 6th International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2012), TEI '12, ACM, pp. 161–168.

[151] MacKenzie, I. S. Fitts' law as a research and design tool in human-computer interaction. *Human Computer Interaction 7*, 1 (Mar. 1992), 91–139.

[152] Maglio, P. P. *The computational basis of interactive skill.* PhD thesis, University of California, San Diego, 1995.

[153] Maglio, P. P., and Kirsh, D. Epistemic action increases with skill. In *Proceedings of the 18th Annual Conference of the Cognitive Science Society* (1996), vol. 16, pp. 391–396.

[154] Maglio, P. P., and Wenger, M. J. Two views are better than one: Epistemic actions may prime. In *In Proceedings of the 22nd Annual Conference of the Cognitive Science Society* (2000).

[155] Maglio, P. P., Wenger, M. J., and Copeland, A. M. The benefits of epistemic action outweigh the costs. In *Proceedings of the 25th annual Conference of the Cognitive Science Society* (2003), pp. 752–757.

[156] Maglio, P. P., Wenger, M. J., and Copeland, A. M. Evidence for the role of self-priming in epistemic action: Expertise and the effective use of memory. *Acta Psychologica 127*, 1 (2008), 72 – 88.

[157] Maher, M. L., and Kim, M. J. Do tangible user interfaces impact spatial cognition in collaborative design? In *Proceedings of the 2nd International Conference on Cooperative Design, Visualization, and Engineering* (Berlin, Heidelberg, 2005), CDVE'05, Springer-Verlag, pp. 30–41.

[158] Marco, J., Baldassarri, S., and Cerezo, E. Toyvision: A toolkit to support the creation of innovative board-games with tangible interaction. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2013), TEI '13, ACM, pp. 291–298.

[159] Marco, J., Cerezo, E., Baldassarri, S., Mazzone, E., and Read, J. C. Bringing tabletop technologies to kindergarten children. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology* (Swinton, UK, UK, 2009), BCS-HCI '09, British Computer Society, pp. 103–111.

[160] Matsushita, N., and Rekimoto, J. Holowall: Designing a finger, hand, body, and object sensitive wall. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1997), UIST '97, ACM, pp. 209–210.

[161] Mazalek, A., and Van den hoven, E. Framing tangible interaction frameworks. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 23*, 3 (2009), 225–235.

[162] McDonagh, E. M., Whirl-Carrillo, M., Garten, Y., Altman, R. B., and Klein, T. E. From pharmacogenomic knowledge acquisition to clinical applications: the PharmGKB as a clinical pharmacogenomic biomarker resource. *Biomarkers in Medicine 5*, 6 (2011), 795–806.

[163] Mehta, N. A flexible machine interface. Master's thesis, University of Toronto, 1982.

[164] Merrill, D., Sun, E., and Kalanithi, J. Sifteo cubes. In *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2012), CHI EA '12, ACM, pp. 1015–1018.

[165] Meyer, S., Cohen, O., and Nilsen, E. Device comparisons for goal-directed drawing tasks. In *Conference Companion on Human Factors in Computing Systems* (New York, NY, USA, 1994), CHI '94, ACM, pp. 251–252.

[166] Microsoft. Printing tags. `https://msdn.microsoft.com/en-us/library/ff727841.aspx`, 2011 (accessed July 7, 2014),.

[167] Moffatt, M. F., Kabesch, M., Liang, L., Dixon, A. L., Strachan, D., Heath, S., Depner, M., von Berg, A., Bufe, A., Rietscchel, E., Heinsmann, A., Simma, B., Frischer, T., Willis-Owen, S. A. G., Wong, K. C. C., Illig, T., Vogelberg, C., Weiland, S. K., von Mutius, E., Abecasis, G. R., Farrall, M., Gut, I. G., Lathrop, G. M., and Cookson, W. O. C. Genetic variants regulating ORMDL3 expression contribute to the risk of childhood asthma. *Nature 448*, 7152 (470-473 2007).

[168] Mueller, M., Goel, A., Thimma, M., Dickens, N. J., Aitman, T. J., and Mangion, J. eQTL explorer: integrated mining of combined genetic linkage and expression experiments. *Bioinformatics 22*, 4 (2006), 509–511.

[169] Nagel, T., Heidmann, F., Condotta, M., and Duval, E. Venice unfolding: A tangible user interface for exploring faceted data in a geographical context. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* (New York, NY, USA, 2010), NordiCHI '10, ACM, pp. 743–746.

[170] Nazari Shirehjini, A., Yassine, A., and Shirmohammadi, S. An rfid-based position and orientation measurement system for mobile objects in intelligent environments. *IEEE Transactions on Instrumentation and Measurement 61*, 6 (2012), 1664–1675.

[171] Neth, H., and Payne, S. J. Interactive coin addition: How hands can help us think. In *Proceedings of the 33rd Annual Conference of the Cognitive Science* (2011), pp. 279–284.

[172] Nica, A. C., and Dermitzakis, E. T. Expression quantitative trait loci: present and future. *Philosophical Transactions of the Royal Society of London B: Biological Sciences 368*, 1620 (2013).

[173] Nielsen, J. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1994), CHI '94, ACM, pp. 152–158.

[174] Nielsen, J. Heuristic evaluation. *Usability Inspection Methods 17*, 1 (1994), 25–62.

[175] Nielsen, J. How to conduct a heuristic evaluation. `https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/`, 1995 (accessed March 9, 2014).

[176] Nokia developer. Asha UX checklist. `http://developer.nokia.com/resources/library/Asha_UI/checklists/asha-ux-checklist.html`, 2016 (accessed March 11, 2016).

[177] Norman, D. A. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine.* Addison-Wesley Longman Publishing, Boston, MA, USA, 1993.

[178] Norman, D. A. *The design of everyday things: Revised and expanded edition.* Basic books, 2013.

[179] Oppenheim, A. N. *Questionnaire design, interviewing and attitude measurement.* Bloomsbury Publishing, 2000.

[180] Ostroff, D., and Shneiderman, B. Selection devices for users of an electronic encyclopedia: An empirical comparison of four possibilities. *Information Processing and Management: an International Journal 24*, 6 (1988), 665–680.

[181] Owen, R., Kurtenbach, G., Fitzmaurice, G., Baudel, T., and Buxton, B. When it gets more difficult, use both hands: Exploring bimanual curve manipulation. In *Proceedings of Graphics Interface* (Waterloo, Ontario, Canada, 2005), GI '05, Canadian Human-Computer Communications Society, pp. 17–24.

[182] OxyPlot contributors. OxyPlot. `http://oxyplot.org/`, 2010 (accessed February 2, 2014).

[183] Patten, J., and Ishii, H. A comparison of spatial organization strategies in graphical and tangible user interfaces. In *Proceedings of Designing Augmented Reality Environments* (New York, NY, USA, 2000), DARE '00, ACM, pp. 41–50.

[184] Patten, J., Ishii, H., Hines, J., and Pangaro, G. Sensetable: A wireless object tracking platform for tangible user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2001), CHI '01, ACM, pp. 253–260.

[185] Patten, J., Recht, B., and Ishii, H. Audiopad: A tag-based interface for musical performance. In *Proceedings of the New Interfaces for Musical Expression* (Singapore, Singapore, 2002), NIME '02, National University of Singapore, pp. 1–6.

[186] Petrelli, D., Dulake, N., Marshall, M., Willox, M., Caparrelli, F., and Goldberg, R. Prototyping tangibles: Exploring form and interaction. In *Proceedings of the 8th International Conference on Tangible,*

Embedded and Embodied Interaction (New York, NY, USA, 2013), TEI '14, ACM, pp. 41–48.

[187] Piper, B., Ratti, C., and Ishii, H. Illuminating clay: A 3-D tangible interface for landscape analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2002), CHI '02, ACM, pp. 355–362.

[188] Pittarello, F., and Stecca, R. Mapping physical objects to digital functions: A tangible interface for querying and navigating a multimedia database. In *Proceedings of the 22nd International Workshop on Database and Expert Systems Applications* (Washington, DC, USA, 2011), DEXA '11, IEEE Computer Society, pp. 134–138.

[189] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., and Carey, T. *Human-Computer Interaction.* Addison-Wesley Longman Ltd., 1994.

[190] Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., De Bakker, P. I., Daly, M. J., et al. PLINK: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics 81*, 3 (2007), 559–575.

[191] Rädle, R., Weiler, A., Huber, S., Jetter, H.-C., Mansmann, S., Reiterer, H., and Scholl, M. H. ebook meets tabletop: Using collaborative visualization for search and serendipity in on-line book repositories. In *Proceedings of the 5th ACM Workshop on Research Advances in Large Digital Book Repositories and Complementary Media* (New York, NY, USA, 2012), BooksOnline '12, ACM, pp. 3–6.

[192] Raffle, H. S., Parkes, A. J., and Ishii, H. Topobo: A constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2004), CHI '04, ACM, pp. 647–654.

[193] Rice, A. C., Harle, R. K., and Beresford, A. R. Analysing fundamental properties of marker-based vision system designs. In *Proceedings of*

the IEEE International Conference on Pervasive Computing and Communications (2006), PerCom '06, pp. 453–471.

[194] Richards, J. E., and Hawley, R. S. *The Human Genome: A User's Guide*, 3rd ed. Academic Press Inc., 2010.

[195] Riemann, J., Khalilbeigi, M., Dezfuli, N., and Mühlhäuser, M. Stacktop: Hybrid physical-digital stacking on interactive tabletops. In *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2015), CHI EA '15, ACM, pp. 1127–1132.

[196] Rockman, M. V., and Kruglyak, L. Genetics of global gene expression. *Nature Reviews Genetics 7*, 11 (2006), 862–872.

[197] Rooney, C., and Ruddle, R. A. A new method for interacting with multi-window application on large, hight resolution displays. In *Proceedings of the 6th Theory and Practice of Computer Graphics Conference* (2008), TPCG '08, pp. 75–82.

[198] Rowberg, J. I2Cdevlib library for MPU6050 6 axis accelerometer/gyroscope. `http://www.i2cdevlib.com/devices/mpu6050`, 2011 (accessed January 11, 2013).

[199] Rzeszotarski, J. M., and Kittur, A. Kinetica: Naturalistic multi-touch data visualization. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), CHI '14, ACM, pp. 897–906.

[200] Sadana, R., and Stasko, J. Designing and implementing an interactive scatterplot visualization for a tablet computer. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2014), AVI '14, ACM, pp. 265–272.

[201] Saffer, D. *Designing Gestural Interfaces: Touchscreens and Interactive Devices.* O'Reilly Media, 2008.

[202] Sasangohar, F., MacKenzie, I. S., and Scott, S. D. Evaluation of mouse and touch input for a tabletop display using fitts' reciprocal tapping task. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (2009), vol. 53, SAGE Publications, pp. 839–843.

[203] Schkolne, S., Ishii, H., and Schroder, P. Immersive design of dma molecules with a tangible interface. In *Visualization, 2004. IEEE* (Oct 2004), pp. 227–234.

[204] Schneider, B., Strait, M., Muller, L., Elfenbein, S., Shaer, O., and Shen, C. Phylo-genie: Engaging students in collaborative 'tree-thinking' through tabletop techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), CHI '12, ACM, pp. 3071–3080.

[205] Schneider, B., Wallace, J., Blikstein, P., and Pea, R. Preparing for future learning with a tangible user interface: The case of neuroscience. *IEEE Transactions on Learning Technologies 6*, 2 (2013), 117–129.

[206] Schöning, J., Brandl, P., Daiber, F., Echtler, F., Hilliges, O., Hook, J., Löchtefeld, M., Motamedi, N., Muller, L., Olivier, P., and Others. Multi-Touch Surfaces: A Technical Guide. Tech. rep., Technical Report TUMI0833: Technical Reports of the Technical University of Munich, 2008.

[207] Schweikardt, E., and Gross, M. D. A brief survey of distributed computational toys. In *Proceedings of the The 1st IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning* (Washington, DC, USA, 2007), DIGITEL '07, IEEE Computer Society, pp. 57–64.

[208] Sears, A., and Shneiderman, B. High precision touchscreens: Design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies 34*, 4 (1991), 593–613.

[209] Sementille, A. C., Lourenço, L. E., Brega, J. R. F., and Rodello, I. A motion capture system using passive markers. In *Proceedings of the ACM SIGGRAPH International Conference on Virtual Reality Continuum*

and Its Applications in Industry (New York, NY, USA, 2004), VRCAI '04, ACM, pp. 440–447.

[210] Shaer, O., and Hornecker, E. Tangible user interfaces: Past, present, and future directions. *Foundations and Trends in Human-Computer Interaction 3*, 1–2 (2010), 1–137.

[211] Shaer, O., Kol, G., Strait, M., Fan, C., Grevet, C., and Elfenbein, S. G-nome surfer: A tabletop interface for collaborative exploration of genomic data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, ACM, pp. 1427–1436.

[212] Shaer, O., Leland, N., Calvillo-Gamez, E. H., and Jacob, R. J. K. The TAC paradigm: Specifying tangible user interfaces. *Personal Ubiquitous Computing 8*, 5 (2004), 359–369.

[213] Shaer, O., Mazalek, A., Ullmer, B., and Konkel, M. From big data to insights: Opportunities and challenges for tei in genomics. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2013), TEI '13, ACM, pp. 109–116.

[214] Shaer, O., Strait, M., Valdes, C., Feng, T., Lintz, M., and Wang, H. Enhancing genomic learning through tabletop interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, ACM, pp. 2817–2826.

[215] Shaer, O., Strait, M., Valdes, C., Wang, H., Feng, T., Lintz, M., Ferreirae, M., Grote, C., Tempel, K., and Liu, S. The design, development, and deployment of a tabletop interface for collaborative exploration of genomic data. *International Journal of Human-Computer Studies 70*, 10 (2012), 746–764.

[216] Sheridan, J. G., and Kortuem, G. Affordance-based design of physical interfaces for ubiquitous environments. In *Ubiquitous Computing Systems*. Springer, 2006, pp. 183–199.

[217] Sherry, S. T., Ward, M.-H., Kholodov, M., Baker, J., Phan, L., Smigielski, E. M., and Sirotkin, K. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research 29*, 1 (2001), 308–311.

[218] Shimazaki, H., and Shinomoto, S. A method for selecting the bin size of a time histogram. *Neural Computing 19*, 6 (2007), 1503–1527.

[219] Shneiderman, B. Direct manipulation: A step beyond programming languages. *Computer 16*, 8 (Aug. 1983), 57–69.

[220] Shneiderman, B. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages* (Washington, DC, USA, 1996), VL '96, IEEE Computer Society, pp. 336–343.

[221] Shneiderman, B., and Maes, P. Direct manipulation vs. interface agents. *Interactions 4*, 6 (1997), 42–61.

[222] Slatkin, M. Linkage disequilibrium - understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics 9*, 6 (2008), 477–485.

[223] Smith, A., and Laine, T. An intelligent fractions learning system: implementation. In *Proceedings of the IST-Africa Conference* (2011), pp. 1–9.

[224] Speicher, M. R., Antonarakis, S. E., and Motulsky, A. G., Eds. *Vogel and Motulsky's Human Genetics: Problems and Approaches*, 4th ed. Springer, 2010, ch. 2.

[225] Spindler, M., Büschel, W., and Dachselt, R. Use your head: Tangible windows for 3D information spaces in a tabletop environment. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2012), ITS '12, ACM, pp. 245–254.

[226] Spindler, M., Büschel, W., Winkler, C., and Dachselt, R. Tangible displays for the masses: spatial interaction with handheld displays by using consumer depth cameras. *Personal and Ubiquitous Computing 18*, 5 (2014), 1213–1225.

[227] Spindler, M., Tominski, C., Schumann, H., and Dachselt, R. Tangible views for information visualization. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2010), ITS '10, ACM, pp. 157–166.

[228] Stone, D., Jarrett, C., Woodroffe, M., and Minocha, S. *User Interface Design and Evaluation (The Morgan Kaufmann Series in Interactive Technologies).* Morgan Kaufmann, San Francisco, CA, USA, 2005.

[229] Stranger, B. E., Nica, A. C., Forrest, M. S., Dimas, A., Bird, C. P., Beazley, C., Ingle, C. E., Dunning, M., Flicek, P., Koller, D., et al. Population genomics of human gene expression. *Nature genetics 39*, 10 (2007), 1217–1224.

[230] Sturtevant, A. H. *A History of Genetics.* Cold Spring Harbor Laboratory Press, 2001.

[231] Szigeti, S. J., Stevens, A., Tu, R., Jofre, A., Gebhardt, A., Chevalier, F., Lee, J., and Diamond, S. L. Output to input: Concepts for physical data representations and tactile user interfaces. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2014), CHI EA '14, ACM, pp. 1813–1818.

[232] Taher, F., Hardy, J., Karnik, A., Weichel, C., Jansen, Y., Hornbæk, K., and Alexander, J. Exploring interactions with physically dynamic bar charts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2015), CHI '15, ACM, pp. 3237–3246.

[233] Tanksley, S. D. Mapping polygenes. *Annual Review of Genetics 27*, 1 (1993), 205–233.

[234] Terrenghi, L., Kirk, D., Richter, H., Krämer, S., Hilliges, O., and Butz, A. Physical handles at the interactive surface: exploring tangibility and its benefits. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2008), ACM, pp. 138–145.

[235] Terrenghi, L., Kirk, D., Sellen, A., and Izadi, S. Affordances for manipulation of physical versus digital media on interactive surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), CHI '07, ACM, pp. 1157–1166.

[236] Terrenghi, L., Kranz, M., Holleis, P., and Schmidt, A. A cube to learn: A tangible user interface for the design of a learning appliance. *Personal Ubiquitous Computng 10*, 2-3 (2006), 153–158.

[237] Ting, S., Kwok, S. K., Tsang, A. H., and Ho, G. T. The study on using passive RFID tags for indoor positioning. *International Journal of Engineering Business Management 3*, 1. (2011), 9–15.

[238] Tobias, E., Maquil, V., and Latour, T. Tulip: A widget-based software framework for tangible tabletop interfaces. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (New York, NY, USA, 2015), EICS '15, ACM, pp. 216–221.

[239] Tominski, C., Gladisch, S., Kister, U., Dachselt, R., and Schumann, H. A survey on interactive lenses in visualization. *EuroVis State-of-the-Art Reports* (2014), 43–62.

[240] Tuddenham, P., Kirk, D., and Izadi, S. Graspables revisited: Multi-touch vs. tangible input for tabletop displays in acquisition and manipulation tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, ACM, pp. 2223–2232.

[241] Ullmer, B., Dell, C., Gil, C., Toole, C., Wiley, C., Dever, Z., Rogge, L., Bradford, R., Riviere, G., Sankaran, R., Liu, K., Freeman, C., Wallace, A., Delatin, M., Washington, C., Reeser, A., Branton, C., and Parker, R. Casier: Structures for composing tangibles and complementary interactors for use across diverse systems. In *Proceedings of the 5th International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2011), TEI '11, pp. 229–236.

[242] Ullmer, B., and Ishii, H. The metaDESK: Models and prototypes for tangible user interfaces. In *Proceedings of the 10th Annual ACM Symposium*

on User Interface Software and Technology (New York, NY, USA, 1997), UIST '97, ACM, pp. 223–232.

[243] Ullmer, B., and Ishii, H. mediaBlocks: Tangible interfaces for online media. In *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 1999), CHI EA '99, ACM, pp. 31–32.

[244] Ullmer, B., and Ishii, H. Emerging frameworks for tangible user interfaces. *IBM Systems Journal 39*, 3-4 (2000), 915–931.

[245] Ullmer, B., Ishii, H., and Jacob, R. J. Tangible query interfaces: Physically constrained tokens for manipulating database queries. In *Proc. of INTERACT* (2003), vol. 3, pp. 279–286.

[246] Ullmer, B., Ishii, H., and Jacob, R. J. K. Token+constraint systems for tangible interaction with digital information. *ACM Transactions on Computer-Human Interaction 12*, 1 (2005), 81–118.

[247] Ullmer, B. A. *Tangible Interfaces for Manipulating Aggregates of Digital Information.* PhD thesis, Cambridge, MA, USA, 2002. AAI0804673.

[248] Underkoffler, J. *The I/O bulb and the luminous room.* PhD thesis, Citeseer, 1999.

[249] Underkoffler, J., and Ishii, H. Illuminating light: An optical design tool with a luminous-tangible interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1998), CHI '98, ACM, pp. 542–549.

[250] Underkoffler, J., and Ishii, H. Urp: A luminous-tangible workbench for urban planning and design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1999), CHI '99, ACM, pp. 386–393.

[251] U.S. Department of Energy Human Genome Project. The science behind the human genome project. `http://www.ornl.gov/sci/techresources/Human_Genome/project/info.shtml`, 2011 (accessed August 12, 2012).

[252] Uttal, D. H., Scudder, K. V., and DeLoache, J. S. Manipulatives as symbols: A new perspective on the use of concrete objects to teach mathematics. *Journal of Applied Developmental Psychology 18*, 1 (1997), 37 – 54.

[253] Valdes, C., Eastman, D., Grote, C., Thatte, S., Shaer, O., Mazalek, A., Ullmer, B., and Konkel, M. K. Exploring the design space of gestural interaction with active tokens through user-defined gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), CHI '14, ACM, pp. 4107–4116.

[254] van Dam, A. Post-WIMP user interfaces. *Communications of the ACM 40*, 2 (1997), 63–67.

[255] Vorst, P., Schneegans, S., Yang, B., and Zell, A. Self-localization with RFID snapshots in densely tagged environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2008), IROS '08, IEEE Computer Society, pp. 1353–1358.

[256] Wehrend, S., and Lewis, C. A problem-oriented classification of visualization techniques. In *Proceedings of the 1st Conference on Visualization '90* (Los Alamitos, CA, USA, 1990), VIS '90, IEEE Computer Society, pp. 139–143.

[257] Weiser, M. The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review 3*, 3 (1999), 3–11.

[258] Welford, A. T. Fundamentals of skill.

[259] Wellner, P. The DigitalDesk calculator: tangible manipulation on a desk top display. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1991), UIST '91, ACM, pp. 27–33.

[260] Welsh, J. B., Zarrinkar, P. P., Sapinoso, L. M., Kern, S. G., Behling, C. A., Monk, B. J., Lockhart, D. J., Burger, R. A., and Hampton, G. M. Analysis of gene expression profiles in normal and neoplastic ovarian tissue samples identifies candidate molecular markers of

epithelial ovarian cancer. *Proceedings of the National Academy of Sciences 98*, 3 (2001), 1176–1181.

[261] Williams, C., Yang, X. D., Partridge, G., Millar-Usiskin, J., Major, A., and Irani, P. Tzee: Exploiting the lighting properties of multi-touch tabletops for tangible 3d interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, ACM, pp. 1363–1372.

[262] Wilson, A. D. PlayAnywhere: a compact interactive tabletop projection-vision system. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2005), UIST '05, ACM, pp. 83–92.

[263] Windows phone development centre. Interactions and usability with windows phone. `http://msdn.microsoft.com/en-us/library/windowsphone/design/hh202889(v=vs.105).aspx#BKMK_Touchtargetsandtext`, 2015 (accessed September 8, 2014).

[264] Wobbrock, J. O., Morris, M. R., and Wilson, A. D. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI '09, ACM, pp. 1083–1092.

[265] Wright, F. A., Shabalin, A. A., and Rusyn, I. Computational tools for discovery and interpretation of expression quantitative trait loci. *Pharmacogenomics 13*, 3 (2012), 343–352.

[266] Xu, W., Chang, K., Francisco, N., Valdes, C., Kincaid, R., and Shaer, O. From wet lab bench to tangible virtual experiment: Synflo. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (New York, NY, USA, 2013), TEI '13, ACM, pp. 399–400.

[267] Yi, J. S., ah Kang, Y., Stasko, J., and Jacko, J. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1224–1231.

[268] Yin, J., and Liu, H. Lever cursor: A bimanual selection technique for multitouch tabletop. In *Proceedings of the IEEE International Conference on Information and Automation* (June 2010), pp. 105–109.

[269] Yin, R. K. *Applications of Case Study Research.* Sage, 2011.

[270] Zhang, Y., Amin, M. G., and Kaushik, S. Localization and tracking of passive RFID tags based on direction estimation. *International Journal of Antennas and Propagation 2007* (2007).

[271] Zhang, Y., Li, X., and Amin, M. Principles and techniques of RFID positioning. *RFID Systems* (2010), 389–415.

[272] Zou, W., Aylor, D., and Zeng, Z.-B. eQTL viewer: visualizing how sequence variation affects genome-wide transcription. *BMC Bioinformatics 8*, 1 (2007), 7.