

# Extending iSEE

*Kevin Rue-Albrecht, Federico Marini, Charlotte Soneson, and  
Aaron Lun*

*2019-12-04*



# Contents

<b>Preface</b>	<b>5</b>
<b>1 Panel classes</b>	<b>7</b>
1.1 Overview . . . . .	7
1.2 The Panel class . . . . .	7
1.3 The DotPlot and Table panel families . . . . .	8
1.4 The ColumnDotPlot and RowDotPlot panel families . . . . .	9
1.5 Built-in ColumnDotPlot panel classes . . . . .	9
1.6 Built-in RowDotPlot panel classes . . . . .	9
1.7 The ColumnTable and RowTable panel families . . . . .	9
1.8 Built-in ColumnTable panel classes . . . . .	9
1.9 Built-in RowTable panel classes . . . . .	9
1.10 The HeatMapPlot panel class . . . . .	9
<b>2 The iSEE server</b>	<b>11</b>
2.1 The app memory . . . . .	11
2.2 The panel API . . . . .	11



# Preface

The Bioconductor *iSEE* package provides functions for creating an interactive graphical user interface (GUI) using the RStudio *Shiny* package for exploring data stored in *SummarizedExperiment* objects, including row- and column-level metadata (Rue-Albrecht et al., 2018). In this book we describe how to create web-applications that leverage built-in panels and develop new ones.



# Chapter 1

## Panel classes

### 1.1 Overview

The types of panels available to compose an *iSEE* app are defined as a hierarchy of S4 classes.

- Panel
  - DotPlot
    - \* ColumnDotPlot
      - RedDimPlot
      - ColDataPlot
      - FeatAssayPlot
    - \* RowDotPlot
      - RowDataPlot
      - SampAssayPlot
  - Table
    - \* RowTable
      - RowStatTable
    - \* ColumnTable
      - ColStatTable
  - HeatMapPlot

### 1.2 The Panel class

The top-most class is called `Panel`. It is a virtual class that defines the core properties common to any panel - existing or future - that may be displayed in the interface.

---

<b>PanelId</b>	Integer index indicating the $i^{\text{th}}$ panel of a given type.
<b>PanelHeight</b>	Height of the panel, in pixels.
<b>PanelWidth</b>	Width of the panel, an integer value indicating the number of columns to use, from 1 to 12.
<b>SelectBoxOpen</b>	Logical value indicating if the <i>Selection parameters</i> box of the panel is open when the app starts.
<b>SelectByPlot</b>	Encoded name of the panel from which to receive a selection of data points.
<b>SelectMultiType</b>	Keyword indicating the method to deal with multiple incoming selections of data points.
<b>SelectMultiSaved</b>	Integer index indicating a single data point selection to use, among multiple incoming selections.

---

### 1.3 The DotPlot and Table panel families

The **Panel** virtual class is directly derived into two major virtual sub-classes:

- **DotPlot**
- **Table**

Those classes introduce properties that are specific to distinct subsets of panel types.

The **DotPlot** class introduce parameters specific to panels where the output is a **ggplot** object and each row in the data-frame is represented as a point in a plot.

The **Table** class introduce parameters specific to panels where the main output is a data-frame directly displayed as a table in the GUI.

In addition, the **HeatMapPlot** class defines a special panel class that directly extends the **Panel** class, as it introduces a set of parameters distinct from both the **DotPlot** and **Table** panel families. This panel type is described in further details in a separate section below.



## 1.4 The ColumnDotPlot and RowDotPlot panel families

### 1.5 Built-in ColumnDotPlot panel classes

### 1.6 Built-in RowDotPlot panel classes

## 1.7 The ColumnTable and RowTable panel families

### 1.8 Built-in ColumnTable panel classes

### 1.9 Built-in RowTable panel classes

## 1.10 The HeatMapPlot panel class

This type of panel introduces parameters specific to panels where the output is a heat map, with each row representing a feature and each column representing a sample in the `se` object.



## Chapter 2

# The iSEE server

### 2.1 The app memory

The `app memory` is a list of instances of those classes, which keeps track of the order in which individual panels are displayed in the GUI.

### 2.2 The panel API

#### 2.2.1 `.cacheCommonInfo`

Each individual panel (e.g., *RedDimPlot*) and family of panels (e.g., *ColDotPlot*) defines a `.cacheCommonInfo` function.

This function is called for each panel in memory when the app is initialized. It allows the app to efficiently compute a single time common information that only depends on the input `se` object, and may be frequently reused during the runtime of the app.

Following the hierarchy of panel types, each call to the signature caches the information in the `se` object itself, before calling `callNextMethod()` to invoke the parent signature. The top-most signature - for the `Panel` class - returns the `se` object that contains all the cached information.

#### 2.2.2 `.refineParameters`

Each panel defines a `.refineParameters` function.



# Bibliography

Rue-Albrecht, K., Marini, F., Soneson, C., and Lun, A. T. L. (2018). isee:  
Interactive summarizedexperiment explorer. *F1000Res*, 7:741.