Tutorial VT2, ISMB 2022

Wed, July 6, 9:00 – 13:00 CDT

Building Interactive Visualizations of Genomics Data with Gosling

Sehi L'Yi  Trevor Manz  Qianwen Wang  Nils Gehlenborg

Biomedical Informatics, Harvard Medical School
Setup Instructions for Tutorial VT2
"Building Interactive Visualizations of Genomics Data with Gosling"

1. If you don't already have, create a Google account (https://www.google.com/account/about/).
2. Click on the following link to open a Google Colab notebook.
   https://tinyurl.com/ismb-vt2-test-notebook
3. Execute all code blocks.
4. If you can see visualizations rendered, you are all set!
5. If you confront any issues, please send an email to us.
   sehi_lyi@hms.harvard.edu
Instructors

Sehi L’Yi
Postdoctoral research fellow in Biomedical Informatics at Harvard Medical School. Interested in designing, implementing, evaluating interactive data visualization.

@sehi_lyi

Trevor Manz
PhD candidate in the Bioinformatics and Integrative Genomics Program at Harvard Medical School. Focused on the intersection of data visualization, bioinformatics, and open-source software.

@trevmanz

Qianwen Wang
Postdoctoral research fellow in Biomedical Informatics at Harvard Medical School. Interested in designing methods and tools for interactive data visualization.

@WangQianwenToo

Nils Gehlenborg
Associate Professor of Biomedical Informatics at Harvard Medical School. Interested in designing methods and tools for data visualization across the spectrum of biomedical data.

@ngehlenborg
Contact

- **Technical Questions & Feature Requests**
  - Github Issues: https://github.com/gosling-lang/gos/issues

- **General Questions**
  - Sehi L'Yi: sehi_lyi@hms.harvard.edu
  - Trevor Manz: trevor_manz@g.harvard.edu
  - Qianwen Wang: qianwen_wang@hms.harvard.edu
  - Nils Gehlenborg: nils@hms.harvard.edu
If You Have Any Questions During Tutorial

- Please use the "Q&A" chat on JUNO

- In each session, we will also give you opportunities to ask questions
Tutorial Overview

- Total 4 hours w/an introduction and five hands-on sessions

- This is a data visualization-focused tutorial

- You will use Google Colab Notebooks & Python with Gos 🦆
Tutorial Goal: What You Are Expected To Learn

- Learn ways to **author interactive Gosling visualizations** in Google Colab using **Gos** 🦆

- Learn the foundations of genomics data visualization
- Learn the key concepts of the *Grammar of Graphics*
  - Used in many popular visualization tools, e.g., Vega-Lite (Altair), ggplot2
- Gain experience with common data visualization techniques
  - e.g., coordinated interactions, semantic zooming
Schedule (09:00–13:00 CDT)

- (09:00–09:40) 🎭 Introduction (40 min)
- (09:40–12:50) 🤝 Five hands-on sessions (160 min + 2 ☕️ breaks)
  - (09:40–10:30) 🤝 Single Track (50 min) → [Link]
  - (10:30–10:45) ☕️ Coffee Break (15 min)
  - (10:45–11:15) 🤝 Track Alignment & Local Data (30 min) → [Link]
  - (11:15–11:45) 🤝 Semantic Zooming (30 min) → [Link]
  - (11:45–12:00) ☕️ Coffee Break (15 min)
  - (12:00–12:30) 🤝 Coordinated Multiple View (30 min) → [Link]
  - (12:30–12:50) 🤝 Share Your Visualizations (20 min) → This Slide Deck
- (12:50–13:00) 🎭 Closing (10 min)
During Five Hands-on Sessions...

- You will **read, execute, and modify** codes in Google Colab Notebooks

- Ensure that you have a Google account
  - You need the account to use Google Colab notebooks
Introduction (Part 1)

Genomics Data Visualization
Tool Catalogs

Awesome Genome Visualization

GenoCAT

https://cmdcolin.github.io/awesome-genome-visualization

http://genocat.tools

486 Tools

107 Tools
Taxonomies for Genomics Data Visualization

- Goal: create a structured description of genomics visualization techniques
  - only consider data that is visualized in the sequence context, i.e., genomic location is represented in the visualization
  - treat genome as a coordinate system

- Address concerns separately:
  - Data Taxonomy, e.g. what data types can be mapped to the genome?
  - Visualization Taxonomy, e.g., how can the coordinate system be laid out? How can the mapped data be encoded?
  - Task Taxonomy, e.g., what kind of tasks are users trying to address with genomic data?
Taxonomy

Data Taxonomy

<table>
<thead>
<tr>
<th>Feature Sets</th>
<th>Interconnection</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Contiguous</td>
<td>Between</td>
<td></td>
</tr>
</tbody>
</table>

Visualization Taxonomy

<table>
<thead>
<tr>
<th>Coordinate System</th>
<th>Tracks</th>
<th>View Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layout</td>
<td>Encoding</td>
<td>Views</td>
</tr>
<tr>
<td>Partition</td>
<td>Alignment</td>
<td>Scales</td>
</tr>
<tr>
<td>Abstraction</td>
<td></td>
<td>Foci</td>
</tr>
<tr>
<td>Arrangement</td>
<td></td>
<td>One</td>
</tr>
</tbody>
</table>

Task Taxonomy

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>Query</td>
</tr>
<tr>
<td>L Lookup</td>
<td>I Identify</td>
</tr>
<tr>
<td>L Locate</td>
<td>C Compare</td>
</tr>
<tr>
<td>L Browse</td>
<td>S Summarize</td>
</tr>
<tr>
<td>L Explore</td>
<td></td>
</tr>
</tbody>
</table>

Mapping

Nusrat, Harbig & Gehlenborg, 2019
## Taxonomy

### Data Taxonomy

<table>
<thead>
<tr>
<th>Feature Sets</th>
<th>Interconnection</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sparse</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Within</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contiguous</td>
</tr>
</tbody>
</table>

### Visualization Taxonomy

<table>
<thead>
<tr>
<th>Coordinate System</th>
<th>Tracks</th>
<th>View Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layout</td>
<td>Encoding</td>
<td>Views</td>
</tr>
<tr>
<td>Partition</td>
<td>Alignment</td>
<td>Scales</td>
</tr>
<tr>
<td>Abstraction</td>
<td></td>
<td>Foci</td>
</tr>
<tr>
<td>Arrangement</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Task Taxonomy

**Tasks**
- Search
- Query
- Lookup
- Identify
- Locate
- Compare
- Browse
- Summarize
- Explore

**Mapping**

Nusrat, Harbig & Gehlenborg, 2019
**Taxonomy**

**Coordinate System:** defines how genomic coordinates are mapped into the visualization space

<table>
<thead>
<tr>
<th>Data Taxonomy</th>
<th>Visualization Taxonomy</th>
<th>Task Taxonomy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Sets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
<td>Tasks</td>
</tr>
<tr>
<td>Sparse</td>
<td></td>
<td>Search</td>
</tr>
<tr>
<td>Interconnection</td>
<td></td>
<td>Query</td>
</tr>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Within</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contiguous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature Sets</td>
<td></td>
<td>Mapping</td>
</tr>
<tr>
<td>Layout</td>
<td></td>
<td>Feature Sets</td>
</tr>
<tr>
<td>Partition</td>
<td></td>
<td>Positions</td>
</tr>
<tr>
<td>Abstraction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arrangement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordinate System</td>
<td>Tracks</td>
<td>View Configurations</td>
</tr>
<tr>
<td>Encoding</td>
<td></td>
<td>Scales</td>
</tr>
<tr>
<td>Alignment</td>
<td></td>
<td>Foci</td>
</tr>
<tr>
<td>Tracks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>View Configurations</td>
<td></td>
<td>Tasks</td>
</tr>
<tr>
<td>Many</td>
<td></td>
<td>Search</td>
</tr>
<tr>
<td>One</td>
<td></td>
<td>Query</td>
</tr>
</tbody>
</table>

Nusrat, Harbig & Gehlenborg, 2019
**Taxonomy**

**Tracks:** a track roughly corresponds to a dataset that maps categorical, quantitative, etc. data to genomic coordinates

---

**Data Taxonomy**

<table>
<thead>
<tr>
<th>Feature Sets</th>
<th>Type</th>
<th>Interconnection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sparsely</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sparse</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contiguous</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coordinate System</th>
<th>Layout</th>
<th>Partition</th>
<th>Abstraction</th>
<th>Arrangement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**View Configurations**

<table>
<thead>
<tr>
<th>View Configurations</th>
<th>Views</th>
<th>Scales</th>
<th>Foci</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Task Taxonomy**

**Tasks**

- **Search**
- **Query**
  - **Identify**
  - **Locate**
  - **Compare**
  - **Browse**
  - **Explore**
  - **Summarize**

---

Nusrat, Harbig & Gehlenborg, 2019
Taxonomy

View Configurations: a view defines a genomic region that is being visualized and consists of one or more tracks; multiple views can be linked to each other.
# Layout

## Data Taxonomy

<table>
<thead>
<tr>
<th>Feature Sets</th>
<th>Type</th>
<th>Interconnection</th>
<th>None</th>
<th>Sparseness</th>
<th>Within</th>
<th>Contiguity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Visualization Taxonomy

<table>
<thead>
<tr>
<th>Coordinate System</th>
<th>Layout</th>
<th>Partition</th>
<th>Abstraction</th>
<th>Arrangement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tracks</th>
<th>Encoding</th>
<th>Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>View Configurations</th>
<th>Views</th>
<th>Scales</th>
<th>Foci</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task Taxonomy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks</td>
</tr>
<tr>
<td>Search</td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>Lookup</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Sets</td>
</tr>
<tr>
<td>D</td>
</tr>
</tbody>
</table>

Nusrat, Harbig & Gehlenborg, 2019
Layout

EpiViz

MizBee

HIC3D Viewer

Hilbert Curve
Layout

Linear

Circular

3D

Space-Filling
Arrangement

Nusrat, Harbig & Gehlenborg, 2019
Arrangement

SyntenyExplorer

Cinteny

HiGlass
Arrangement

Serial

Parallel

Orthogonal
Track Alignment
Track Alignment
Track Alignment

Parallel
Track Alignment
Views

Data Taxonomy

Feature Sets
- Type
- Interconnection

Visualization Taxonomy

Coordinate System
- Layout
- Partition
- Abstraction
- Arrangement

Tracks
- Encoding
- Alignment

View Configurations
- Views
- Scales
- Foci

Task Taxonomy

Tasks
- Search
- Query
  - L: Lookup
  - I: Identify
  - L: Locate
  - C: Compare
  - B: Browse
  - E: Explore
  - S: Summarize

Mapping
- Feature Sets
  - Positions

Nusrat, Harbig & Gehlenborg, 2019
Views
Scales

Genome

Chromosome

Detail

3 Scales

Genome

1 Scale
Foci

Genome > Chromosome > Detail

1 Focus

IGV
Foci

Genome > Chromosome > Detail

MizBee

1 Focus

Multi Focus
Taxonomy

Data Taxonomy

Feature Sets
- Type: [[Image of symbols representing different types of features]]
- Interconnection: [[Image of symbols representing different levels of interconnection]]

Visualization Taxonomy

Coordinate System
- Layout: [[Image of symbols for different layout styles]]
- Partition: [[Image of symbols for different partition styles]]
- Abstraction: [[Image of symbols for different abstraction styles]]
- Arrangement: [[Image of symbols for different arrangement styles]]

Tracks
- Encoding: [[Image of symbols for different encoding methods]]
- Alignment: [[Image of symbols for different alignment methods]]

View Configurations
- Views: [[Image of symbols for different view styles]]
- Scales: [[Image of symbols for different scale styles]]
- Foci: [[Image of symbols for different focus styles]]

Task Taxonomy

Tasks
- Search: Lookup, Locate
- Query: Identify, Compare, Browse, Explore, Summarize

Mapping
- Feature Sets: B, S, C
- Positions: Lp, I, L, S, C, L, E
Challenges in Genomics Data Visualization

- Everything is connected
  - Different data types need to be integrated: sequence, expression levels, metabolites, phenotype information, etc.
  - Many different types of data need to be loaded into a single software
- Large space with sparse distribution of patterns across multiple scales
- Many types of patterns along the genome (SNP, epigenomic peaks, genomic rearrangements, etc.)
Introduction (Part 2)

The Gosling Grammar
Gosling

- A grammar for scalable and interactive genome-mapped data visualization (L'Yi et al. IEEE TVCG 2022)
Gosling Key Features

- **Expressiveness**
  - Based on a visualization grammar
- **Data scalability**
  - Built on HiGlass (Kerpedjiev et al. Genome Biology 2018)
- **Encoding scalability**
  - Semantic zooming & responsive design
- **Coordinated interactivity**
  - Zooming, panning, brushing, linking, ...
Supported File Formats (as of June 29, 2022)

Plain Datasets (No HiGlass Server)
- JSON
- CSV (TSV, BED, etc)
- BigWig
- BAM (with BAI)
- VCF (with TBI)

Pre-aggregated Datasets (HiGlass Server)
- Vector (1D Quantitative values)
- Multi-Vec (N-D Quantitative values)
- BEDDB (Any BED-like formats)
- Cooler (Matrix)

Details: http://gosling-lang.org/docs/data
Gosling Ecosystem

Gosling.js (JavaScript/TypeScript)
- JSON text → Visualization
- Web application/React
- Online Editor (Link)

Gos (Python)
- Python functions → Visualization
- Jupyter Notebook/Lab & Colab
- Preprint (Manz et al. 2022)
Gosling Ecosystem

Gosling.js (JavaScript/TypeScript)
JSON text → Visualization
Web application/React
Online Editor (https://gosling.js.org)

Gos (Python)
Python functions → Visualization
Jupyter Notebook/Lab & Colab
Preprint (Manz et al. 2022)
The Gosling Grammar: Tracks & Views

- **Track**: The core component of a genomics visualization that defines mappings of genomics data to visual properties

- **View**: A group of tracks that shares the same linked genomic domain
The Gosling Grammar: Tracks & Views

- Gosling Visualization
The Gosling Grammar: Tracks & Views

- Gosling Visualization
  - View 1 (Chr1)
  - View 2 (Chr2)
  - View 3 (Chr3)
The Gosling Grammar: Tracks & Views

- Gosling Visualization
  - View 1 (Chr1)
    - Track 1 (Bar)
    - Track 2 (Ideogram)
  - View 2 (Chr2)
    - Track 3 (Bar)
    - Track 4 (Ideogram)
  - View 3 (Chr3)
    - Track 5 (Bar)
    - Track 6 (Ideogram)
The Gosling Grammar: Tracks & Views

- Gosling JSON Spec

```json
{"views": [
    {"tracks": [ {"..."}, {"..."} ] },
    {"tracks": [ {"..."}, {"..."} ] },
    {"tracks": [ {"..."}, {"..."} ] },
    {"tracks": [ {"..."}, {"..."} ] }
]}
```

- A grammar used to describe and create a wide range of statistical graphics
- Adopted and inspired to many visualization libraries
  - Vega-Lite & Altair
  - ggplot2
- Decomposes visualization into multiple primitives
  - Mark
  - Scale
  - Visual channel
  - Layer
  - ...

Statistics and Computing
Leland Wilkinson
The Grammar of Graphics
Second Edition

Springer
Grammar of Graphics in Gosling: Defining Track

- **Mark**
  - point, bar, rect, line, area, text, withinLink, etc

- **Visual channel**
  - x, y, color, size, stroke, strokeWidth

- **Scale**
  - domain, range
Example: \textit{Point} mark + \textit{Size} channel

\begin{center}
\begin{tabular}{c|c}
\textbf{domain} & \textbf{range (px)} \\
\hline
1 & 12 \\
0 & 2 \\
\end{tabular}
\end{center}
Example: \textit{Point mark + Size channel}

\begin{itemize}
\item Your Data
\item \[ \text{domain} \]
\item \[ \text{range (px)} \]
\end{itemize}

\[ \begin{align*}
\text{domain} & : 1 \\
\text{range (px)} & : 12
\end{align*} \]
Example: *Point* mark + *Size* channel

Your Data

### Domain

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
</table>

### Range (px)

<table>
<thead>
<tr>
<th>0</th>
<th>2</th>
</tr>
</thead>
</table>

0.9

0.2
Example: \textit{Point} mark + \textit{Size} channel

\begin{align*}
\text{domain} & \quad \text{range (px)} \\
1 & \quad 12 \\
0.9 & \\
0.6 & \\
0.2 & \\
0 & \quad 2
\end{align*}
Example: *Point* mark + *Size* channel

Tabularized
(Abstracted)
Genomics Data
All genomics files supported in Gosling (e.g., BigWig, BED, BAM) are internally converted to a tabular representation, having explicit column names. These column names are used to specify visual encoding.

<table>
<thead>
<tr>
<th>chr</th>
<th>position</th>
<th>peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;chr1&quot;</td>
<td>123,000</td>
<td>0.142</td>
</tr>
<tr>
<td>&quot;chr1&quot;</td>
<td>124,000</td>
<td>0.321</td>
</tr>
<tr>
<td>&quot;chr1&quot;</td>
<td>125,000</td>
<td>0.951</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**BigWig**

**Internal Tabular BigWig Data In Gosling**
```json
{
  "data": {
    "url": "https://gosling-lang.org/data/example1.bw",
    "type": "bigwig",
  },
  "mark": "rect",
  "x": {
    "field": "position",
    "type": "genomic"
  },
  "color": {
    "field": "peak",
    "type": "quantitative"
  },
  "width": 400,
  "height": 130
}
```
```json
{
  "data": {
    "url": "https://gosling-lang.org/data/example1.bw",
    "type": "bigwig",
  },
  "mark": "line",
  "x": {
    "field": "position",
    "type": "genomic"
  },
  "y": {
    "field": "peak",
    "type": "quantitative"
  },
  "width": 400,
  "height": 130
}
```
```json
{
  "data": {
    "url": "https://gosling-lang.org/data/example1.bw",
    "type": "bigwig",
    "mark": "point",
    "x": {"field": "position", "type": "genomic" },
    "y": {"field": "peak", "type": "quantitative" },
    "width": 480,
    "height": 130
  }
}
```
{  
  "data": {  
    "url": "https://gosling-lang.org/data/example1.bw",
    "type": "bigwig", ...
  },  
  "mark": "point",
  "x": {"field": "position", "type": "genomic" },
  "y": {"field": "peak", "type": "quantitative" },
  "size": {"field": "peak", "type": "quantitative" },
  "color": {"field": "peak", "type": "quantitative" },
  "width": 400,
  "height": 130
}
Composition Primitives

- **alignment** of tracks (stack, overlay)
- **arrangement** of views (parallel, serial, vertical, horizontal)
In the Following Hands-On Sessions...

- **Session 1: Single Track**
  - Author custom tracks based on grammar of graphics

- **Session 2: Track Alignment & Local Data**
  - Compose tracks either by stacking and overlaying (e.g., genes)

- **Session 3: Semantic Zooming**
  - Use a unique primitive in Gosling to reveal patterns across scales

- **Session 4: Coordinated Multiple View**
  - Compose multiple views with coordinated interactions

- **Session 5: Share Your Visualization**
  - Create shareable web-based applications for your Gosling visualizations
Using Google Colab Notebooks

Copy Notebooks

1. Login using your Google account
2. Save a copy of a notebook in your Drive
3. Use the copied notebook

Be aware of mouse positions for scrolling

- Gos visualizations use mouse wheel for navigation!
Session 1: Single Track

Author custom tracks based on the Grammar of Graphics

Link to Google Colab
Coffee Break (15 min)
Session 2: Track Alignment & Local Data

Compose tracks either by stacking and overlaying (e.g., genes)

Link to Google Colab
Session 3: Semantic Zooming

Use a unique primitive in Gosling to reveal patterns across scales

Link to Google Colab
Coffee Break (15 min)
Session 4: Coordinated Multiple Views

Compose multiple views with coordinated interactions

Link to Google Colab
Session 5: Share Your Visualization

Create shareable web-based applications for your Gosling visualizations

We will use the following slides
Various Ways to Share Visualizations

- A URL link that contains a gosling spec
- A HTML file
- Github Gist + Online Editor

Due to time limits, we do not cover the creation of Streamlit apps and React apps in this tutorial.

- Streamlit App (Python)
  Please refer to Streamlit-gosling repo

- React App (Javascript)
  Please refer to gosling doc and gosling-react repo for more details.
A URL Link

- Step 1: Copy the values from the json file we just downloaded to the online editor.
- Step 2: Click EXPORT->SAVE URL. A URL will be copied to your clipboard!

the orange string in URL is a compression of the gosling JSON spec using JSONCrush

https://gosling.js.org
An HTML File

- Step 1: Copy the values from the json file we just downloaded to the online editor.
- Step 2: Click EXPORT->HTML. A URL will be copied to your clipboard!
An HTML File

- Step 1: Create a HTML file using this template.

- Step 2: Replace `{your_gosling_spec}` with the spec of your Gosling visualization.

```html
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="https://unpkg.com/higlass@1.11/dist/hglib.css ">
  <script src="https://unpkg.com/react@17/umd/react.production.min.js ">
  </script>
  <script src="https://unpkg.com/react-dom@17/umd/react-dom.production.min.js ">
  </script>
  <script src="https://unpkg.com/pixi.js@6/dist/browser/pixi.min.js ">
  </script>
  <script src="https://unpkg.com/higlass@1.11/dist/hglib.js ">
  </script>
  <script src="https://unpkg.com/gosling.js@0.9.17/dist/gosling.js ">
  </script>
</head>
<body>
  <div id="gosling-container "/>
  <script>
    gosling.embed (
      document.getElementById('gosling-container'),
      `{your_gosling_spec}
    );
  </script>
</body>
</html>
```
Github Gist + Gosling Editor

- Step 1: Create a github gist for the json file of your gosling visualization. Ensure that the name of your json file starts with "gosling.js".

- Step 2 (Optional): Add a readme.md file and a title to your gist.

- Step 3: You can then share your visualization through https://gosling.js.org/?gist={github_username}/{gist_id}
Github Gist + Gosling Editor

- Step 1: Create a github gist for the json file of your gosling visualization. Ensure that the name of your json file starts with "gosling.js".

- Step 2 (Optional): Add a readme.md file and a title to your gist.

- Step 3: You can then share your visualization through https://gosling.js.org/?gist={github_username}/{gist_id}

https://gist.github.com/wangqianwen0418/6f092d419feceae20273de919f5a04284

https://gosling.js.org/?gist=wangqianwen0418/6f092d419feceae20273de919f5a04284
Hands-On Exercises

Choose one visualization you created in previous sessions:

- Save the Gosling spec of the visualization
- Create an HTML file for the visualization
- Store the Gosling spec as a Github Gist and open the visualization through the Gosling online editor
Closing
Story of the Name “Gosling”

- Grammar Of Scalable Linked Interactive Nucleotide Graphics
- Baby goose
Story of the Name “Gosling”

- Grammar Of Scalable Linked Interactive Nucleotide Graphics
- Baby goose
- Raymond Gosling
Story of the Name “Gosling”

- Grammar Of Scalable Linked Interactive Nucleotide Graphics
- Baby goose
- Raymond Gosling
Contact

- **Technical Questions & Feature Requests**
  - Github Issues: https://github.com/gosling-lang/gos/issues

- **General Questions**
  - Sehi L'Yi: sehi_lyi@hms.harvard.edu
  - Trevor Manz: trevor_manz@g.harvard.edu
  - Qianwen Wang: qianwen_wang@hms.harvard.edu
  - Nils Gehlenborg: nils@hms.harvard.edu
Resources

- All tutorial materials will be also available at...
  http://gosling-lang.org/tutorials/ismb
- Documentation
  http://gosling-lang.org/
- Source codes (Github)
  https://github.com/gosling-lang
- Online editor
  https://gosling.js.org/
- Papers
  L'Yi et al. Gosling, IEEE TVCG 2022
  Manz et al. Gos, OSF Preprint 2022
ISMB Tutorial Feedback Survey

[Link]
APPENDIX

Applications of Using Gosling
Online Editor (https://gosling.js.org)

A interactive website that enables users to author Gosling.js visualization
A template GitHub repository to use Gosling.js and its API in a React application

demo: https://gosling-lang.github.io/gosling-react/
GenoREC

- A knowledge-based recommendation model & interface for genome-mapped data visualization
  - Gosling.js is used as a rendering engine
  - Preprint: Pandey et al. OSF 2022
Interactive Browser for Somatic Structural Variation

A web-based interactive multi-scale visualization tool that enables browsing individual samples in **multiple levels of scale** from whole genome, chromosome, and individual genes to raw reads

(Demo: https://sehilyi.github.io/gosling-app-sv/)
Interactive Browser for Somatic Structural Variation

A web-based interactive multi-scale visualization tool that enables browsing individual samples in multiple levels of scale from whole genome, chromosome, and individual genes to raw reads

(Demo: https://sehilyi.github.io/gosling-app-sv/)
Interactive Browser for Somatic Structural Variation

A web-based interactive multi-scale visualization tool that enables browsing individual samples in **multiple levels of scale** from whole genome, chromosome, and individual genes to raw reads

(Demo: https://sehilyi.github.io/gosling-app-sv/)