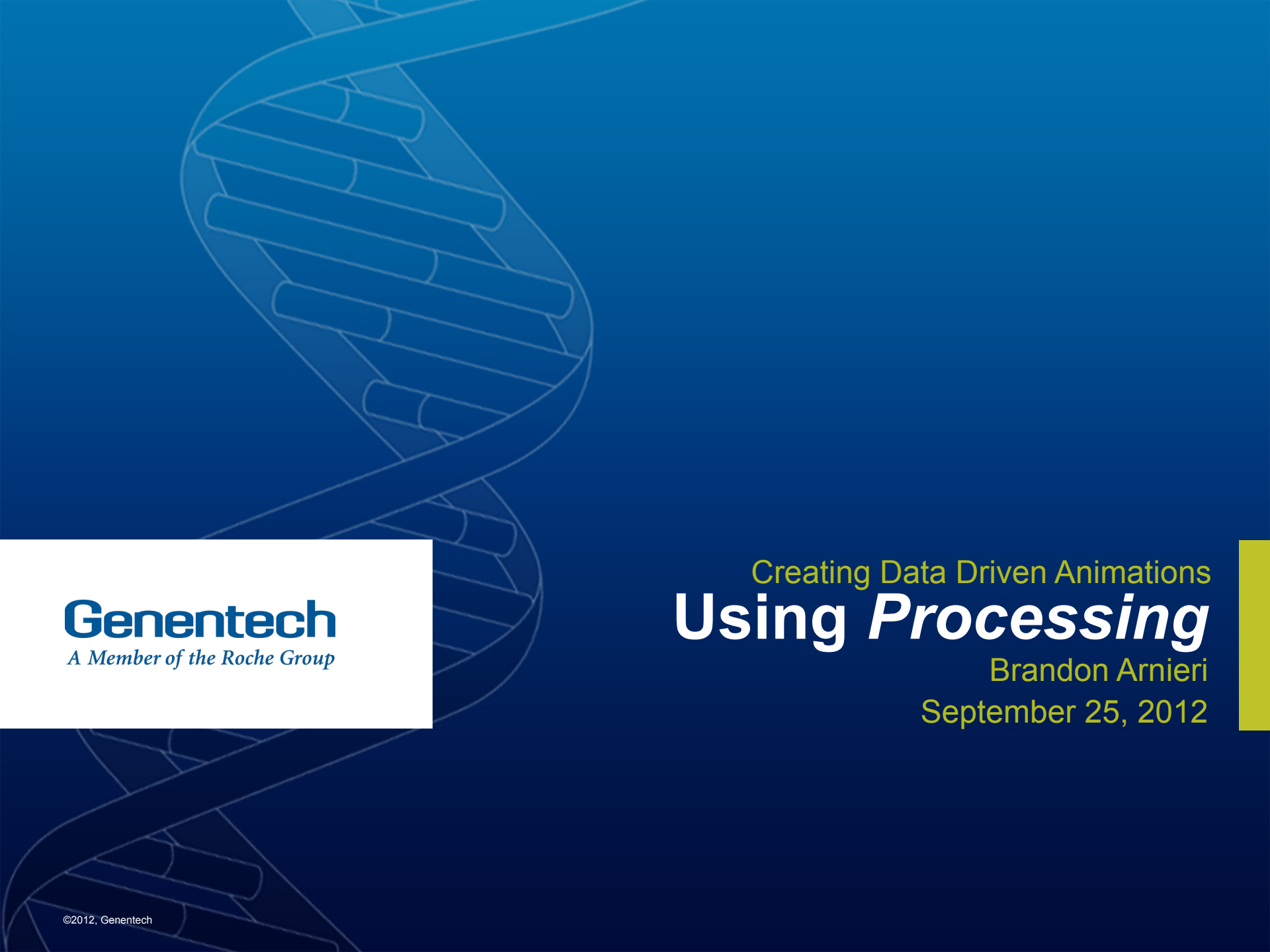




Genentech

A Member of the Roche Group

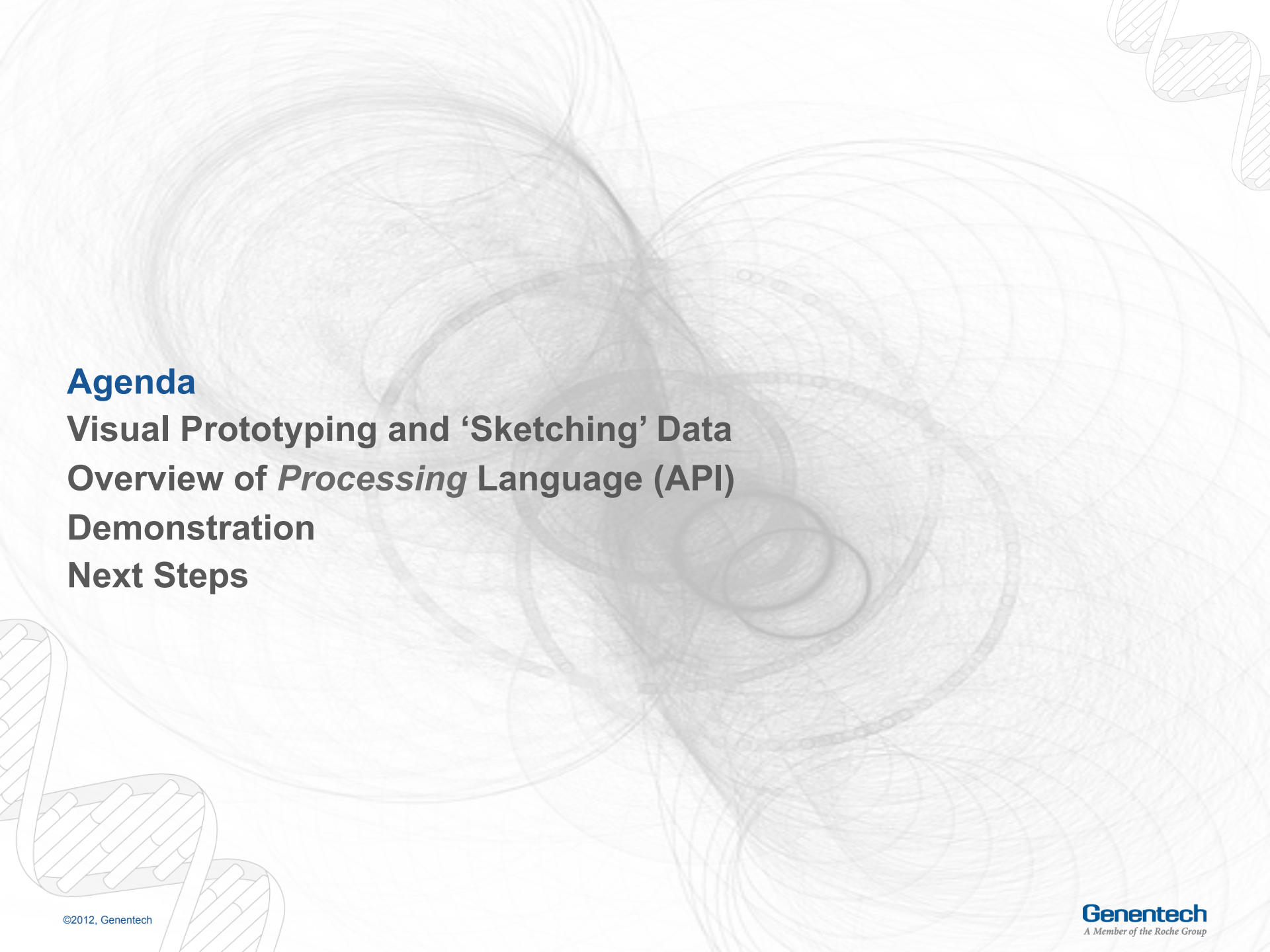
Creating Data Driven Animations
Using *Processing*

Brandon Arnieri
September 25, 2012



Using *Processing* to Create Data Driven Animations

Introduction



Agenda

Visual Prototyping and ‘Sketching’ Data
Overview of *Processing* Language (API)
Demonstration
Next Steps



Visual Prototyping and ‘Sketching’ Ideas

Designing New Tools from Scratch

Visual Prototyping and 'Sketching' Ideas

Putting the patient data puzzle together (CDISC + EHR+ everything else)

Clinicians, scientists, researchers and patients need more visual, interactive tools (exploratory and expository)

Hans Rosling's 2006 TED Talk: No More Boring Data (Gapminder)

Data visualization prototyping and sketching foster creativity and new ideas for detecting patterns

Data art (FlowingData)

Several open-source tools (APIs) for developers and data artists:

- R
- D3.js (Data-Driven Documents)
- Open Flash Chart
- Flare
- OpenLayers
- Google Chart Tools (MotionChart)
- Processing (today's talk is about)



Overview of *Processing* Language (API)

Basics and PDE

Overview of *Processing* Language (API)

Relates software concepts to principles of visual form, motion, and interaction.

Founded by Ben Fry and Casey Reas in 2001 at MIT Media Lab

Easy to write software for drawing, animation, and reacting to the environment (interaction)

Free to download and open source

Well documented, with many books available

Used by tens of thousands of students, artists, designers, researchers and hobbyists for learning, prototyping, and production. Active online community

Core language makes use of Java and elements identical to C

Processing.js enables code to run by any HTML5 compatible browser

Can integrate various media types including audio, video and electronics devices (e.g., Microsoft Kinect, Arduino I/O boards)

Anatomy of a *Processing* Program

Variable name
Data type
Assignment operator
Statement terminator

```
int y = 0;
```

The area between { and } is a block

```
void setup() {  
  size(300, 300);  
}
```

Return value
Function
Parameters
Expression

```
void draw() {  
  line(0, y, 300, y);  
  y = y + 4;  
}
```

Reading and Parsing Your Data in *Processing*

Multiple ways to load data from various locations

- Preprocessed text data (e.g., *.csv*, *.tsv*, *.txt*)
- XML
- Using a database

Several functions for parsing text files, XML, JSON

Preferred method:

1. Create a `Record` class to declare (data types) and create the objects (assign variable names) to load and store each line of data.
2. Use iterative loop to load each line of data as string from arrays in to `Record` object we can call `datalines`.

3. Now, the *i-th* row of a variable is stored in

`datalines[i].<variable name>`



Data-Driven Motion Using *Processing* Demonstration



Genentech
A Member of the Roche Group