Academic Journey Section (Interactive Timeline)

Last Updated: January 13, 2026 **Related Docs:** ABOUT-SECTION.md | HERO-SECTION.md | ARCHITECTURE.md

---

Table of Contents

Overview

The Academic Journey Section is an **interactive timeline** that maps career milestones to protein structures, combining personal narrative with scientific visualization.

Design Philosophy

Proteins as Metaphors: Each academic experience is represented by a protein structure that symbolizes that period's theme:

Period	Institution	Protein	Metaphor
2016-2019	Universidad Europea de Madrid	Histone H1	Lysine “hooks” organizing genome → Foundational knowledge organizing understanding
2019-2022	University of Queensland	GFP (Green Fluorescent Protein)	Making biological processes visible → Coding making data visible
2022-2024	University of Technology Sydney	BRCA1	Cancer risk gene → Human impact of genetic counseling
2022-2025	IGM Team	Cas9	Genome editing tool → Building tools that change systems
2025-Present	Lanzarote & Beyond	RNA Polymerase II	Multi-subunit synthesis → Consolidating skills

Key Features

- **5 Timeline Entries**: 2016 → 2025 (9-year journey)
- **Clickable Cards**: Each card opens a full-screen modal with 3D protein viewer
- **Interactive Proteins**: Toggle structural elements (helices, sheets, lysines)
- **Hover-Linked Text**: Scientific descriptions highlight when hovering protein structures
- **Theme-Aware**: Adapts colors for light/dark modes
- **Mobile Optimized**: Stacked cards on mobile, side-by-side timeline on desktop

Component Architecture

File Structure

```
src/
├── components/
│   └── about/
│     ├── AcademicJourney.jsx      # Main timeline component (1,469 lines)
│     ├── ProteinViewer.jsx       # Three.js protein renderer (500+ lines)
│       └── TimelineSegment.jsx        # Curved line renderer (31 lines)
├── assets/
│   ├── HistoneH1_V2.glb              # Histone H1 protein model
│   ├── GFP_v2.glb                    # Green Fluorescent Protein model
│   ├── BRCA1.glb                     # BRCA1 tumor suppressor
│   ├── Cas9prot.glb                  # Cas9 nuclease
│   └── RNApol2.glb                   # RNA Polymerase II
└── contexts/
    └── ThemeContext.jsx              # Dark/light theme state
```

Component Hierarchy

```
<AcademicJourney>
├── <Modal> (ReactDOM.createPortal)
│   ├── Modal Overlay (backdrop blur)
│   ├── Close Button (×)
│   ├── Left Panel: <ProteinViewer> (interactive)
│   │   ├── Three.js Canvas
│   │   ├── OrbitControls
│   │   └── GLB Model (protein)
│   └── Right Panel: Content + Controls
│       ├── Scientific Section (protein description)
│       ├── Control Center (toggle buttons)
│       │   ├── Surface Toggle
│       │   ├── Lysines Toggle
│       │   ├── Helices Toggle
│       │   └── Sheets Toggle
│       ├── Hover Info Display
│       └── Personal Journey Section
│
└── <section className="academic-journey-wrapper">
    ├── Timeline Intro Block
    ├── Timeline Dot (year) × 5
    ├── Timeline Card × 5
    │   ├── <ProteinViewer> (preview, non-interactive)
    │   └── Card Text (title, years, description)
    └── Laser Connectors × 4 (desktop only)
```

State Management

11 State Variables:

```
const [selectedExperience, setSelectedExperience] = useState(null); // Modal state
const [isMobile, setIsMobile] = useState(false);                    // Mobile detection
const [hoveredStructure, setHoveredStructure] = useState(null);     // Protein hover state

// Generic controls
const [showSurface, setShowSurface] = useState(true);

// Histone-specific controls
const [showLysines, setShowLysines] = useState(false);

// GFP-specific controls
const [showBarrel, setShowBarrel] = useState(true);
const [showChromophore, setShowChromophore] = useState(true);
const [showInteractions, setShowInteractions] = useState(false);

// General structural controls
const [showHelices, setShowHelices] = useState(true);
const [showSheets, setShowSheets] = useState(true);
```

Experience Data Structure

experiences Array

5 objects representing career milestones:

```
const experiences = [
  {
    id: 1,
    title: "Universidad Europea de Madrid",
    years: "2016–2019",
    description: "Forging a first-principles toolkit for biology.",
    expandedContent: {
      scientific: {
        proteinName: "Histone H1",
        description: "Histone H1 is a linker histone that acts as the genome's librarian..."
      },
      personal: {
        whatIDid: "This is where I built the frame for everything that followed...",
        whyItMattered: "The discipline forged in math and physics was worth the grind..."
      }
    },
    component: (
      <ProteinViewer
        path={histoneH1}
        position={[0, 0, 0]}
        scale={[1.0, 1.0, 1.0]}
        cameraZ={40}
        tooltip="Histone H1 (PDB: 1HST)"
      />
    )
  },
  // ... 4 more experiences
];
```

Data Fields

Field	Type	Purpose
id	number	Unique identifier (1-5)
title	string	Institution name
years	string	Time period (e.g., "2016–2019")
description	string	Short tagline for card
expandedContent	object	Modal content (scientific + personal)
component	JSX	`<ProteinViewer>` component with props

expandedContent Structure

```
expandedContent: {
  scientific: {
    proteinName: "Histone H1",
    description: "Detailed scientific explanation with structure terms..."
  },
  personal: {
    whatIDid: "First-person narrative of activities...",
    whyItMattered: "Reflection on significance..."
  }
}
```

Why Two Sections? - **Scientific**: Protein biology (appeals to technical audience) - **Personal**: Career narrative (appeals to all audiences)

Timeline Layout System

Desktop Layout (> 768px)

Absolute Positioning with CSS custom properties:

```
--timeline-side-offset: 12%;    /* Distance from viewport edge */
--timeline-card-width: 36%;     /* Card width */
```

Visual Structure:

```
|<--12%-->|<-- Card 36% -->|<-- Center -->|<-- Card 36% -->|<--12%-->|
|          |  Universidad   | === Dot === |                |          |
|          |  Europea       | === 2016 == |                |          |
|          |  (right-card)  |             |                |          |
|          |                |             |                |          |
|          |                | === Laser ===                |          |
|          |                |     Line    |                |          |
|          |                |             |                |          |
|          |                | === Dot === |  University of |          |
|          |                | === 2019 == |  Queensland    |          |
|          |                |             |  (left-card)   |          |
```

Positioning Logic:

```
// Right-side cards
style={{
  position: 'absolute',
  top: '50px',                           // Y-position
  right: 'var(--timeline-side-offset)',   // 12% from right
  width: 'var(--timeline-card-width)',    // 36% width
  zIndex: 5
}}

// Left-side cards
style={{
  position: 'absolute',
  top: '600px',                          // Y-position
  left: 'var(--timeline-side-offset)',    // 12% from left
  width: 'var(--timeline-card-width)',    // 36% width
  zIndex: 5
}}

// Timeline dots
style={{
  position: 'absolute',
  top: '50px',                           // Y-position
  left: 'calc(50% - 30px)',               // Centered (dot width = 60px)
  zIndex: 10
}}
```

Mobile Layout (768px)

Static Stacked Layout:

```
style={{
  position: 'static',           // No absolute positioning
  width: '90%',                // Responsive width
  maxWidth: '600px',           // Cap at 600px
  margin: '0 auto 2.5rem auto', // Centered with bottom margin
  left: 'unset',
  right: 'unset',
  top: 'unset'
}}
```

Visual Structure:

```
┌──────────────────────┐
│  Timeline Intro      │
│  (text block)        │
└──────────────────────┘


┌──────────────────────┐
│   2016 Dot           │
└──────────────────────┘
┌──────────────────────┐
│  Card 1              │
│  Universidad         │
│  [Protein Preview]   │
└──────────────────────┘


┌──────────────────────┐
│   2019 Dot           │
└──────────────────────┘
┌──────────────────────┐
```

```
│  Card 2             │
│  University of      │
│  Queensland         │
│  [Protein Preview]  │
└─────────────────────┘
```

...

Why Stacked? - **Readability**: Narrow viewports can't fit side-by-side cards - **Natural Flow**: Vertical scrolling follows chronological order - **Simplified Animations**: No complex absolute positioning

Laser Connectors

Desktop Only (hidden on mobile):

```
{!isMobile && (
  <motion.div
    initial={{ height: 0 }}
    whileInView={{ height: 534 }}
    viewport={{ once: true, amount: 0.5 }}
    transition={{ duration: 1.2, ease: "easeInOut" }}
    style={{
      position: 'absolute',
      top: '66px',
      left: 'calc(50% - 1px)',
      width: '2px',
      background: isDark
        ? 'linear-gradient(to bottom, rgba(255, 255, 255, 0.35), rgba(200, 200, 200, 0.25))'
        : 'linear-gradient(to bottom, #ffffff, #fef3cd, #fed7aa, #fb923c)',
      boxShadow: isDark
        ? '0 0 4px rgba(255, 255, 255, 0.2), 0 0 8px rgba(255, 255, 255, 0.1)'
        : '0 0 12px rgba(255, 248, 220, 0.9), 0 0 24px rgba(251, 146, 60, 0.7)',
      opacity: isDark ? 0.35 : 1,
      zIndex: 8
    }}
  />
)}
```

Visual Effect: - **Animated Growth**: Line height animates from 0 → 534px on scroll into view - **Gradient**: White → Orange gradient (light mode), subtle white (dark mode) - **Box Shadow**: Glowing effect in light mode, subtle in dark mode - **Purpose**: Visual connector between timeline dots

Four Connectors: 1. 2016 → 2019 (height: 534px) 2. 2019 → 2022 (height: 534px) 3. 2022 → 2022 (between UTS and IGM, height: 784px) 4. 2022 → 2025 (height varies)

Interactive Modal System

Modal Architecture

ReactDOM.createPortal:

```
const modalContent = selectedExperience && ReactDOM.createPortal(
  (
    <>
      <div className="modal-overlay" onClick={closeModal} />
      <div className="modal-content">
        {/* Modal content */}
      </div>
    </>
  ),
  document.body
);
```

Why Portals? - **DOM Hierarchy**: Renders modal outside React component tree - **Z-Index Freedom**: Avoids z-index conflicts with parent containers - **Accessibility**: Direct child of `<body>` for screen readers

Opening Modal

Click Handler:

```
const handleCardClick = (experience) => {
  const scrollY = window.scrollY;
  document.body.style.top = `-${scrollY}px`;
  document.body.classList.add('prevent-scroll');
  document.body.classList.add('modal-active');
  setSelectedExperience(experience);
};
```

Scroll Locking Steps: 1. **Save Scroll Position**: `const scrollY = window.scrollY` 2. **Fix Body Position**: `document.body.style.top = '-${scrollY}px'` 3. **Add Prevent-Scroll Class**: CSS rule `overflow: hidden`

4. **Add Modal-Active Class**: Additional styling for modal state 5. **Set Selected Experience**: Triggers modal render

Why This Pattern? - **iOS Safari Bug**: Simple `overflow: hidden` doesn't work on iOS - **Preserve Position**: User returns to exact scroll position on close - **No Jump**: Prevents scroll jump when body becomes fixed

Closing Modal

Close Triggers: 1. **Click Overlay**: `<div className="modal-overlay" onClick={closeModal} />` 2. **Click Close Button**: `<button onClick={closeModal}>×</button>` 3. **Press Escape**: `document.addEventListener('keydown', handleEscape)`

Close Handler:

```
const closeModal = () => {
  setSelectedExperience(null);
};

useEffect(() => {
  if (!selectedExperience) {
    const scrollY = document.body.style.top;
    document.body.classList.remove('prevent-scroll');
    document.body.classList.remove('modal-active');
    document.body.style.top = '';
    if (scrollY) {
      window.scrollTo(0, parseInt(scrollY || '0') * -1);
    }
  }
}, [selectedExperience]);
```

Scroll Restoration Steps: 1. **Read Saved Position**: `const scrollY = document.body.style.top` 2. **Remove Classes**: `prevent-scroll`, `modal-active` 3. **Clear Top Style**: `document.body.style.top = ''` 4. **Restore Scroll**: `window.scrollTo(0, parseInt(scrollY) * -1)`

Why useEffect? - **Cleanup**: Runs on every `selectedExperience` change - **Guaranteed Execution**: Ensures scroll always restored - **Return Cleanup**: Also clears on component unmount

Modal Layout

Split-Panel Design:

```
┌───────────────────────────────────────────┐
```

```
│                                    [×]  │ Close button (top-right)
│  ┌────────────┬────────────────────────┐ │
│  │            │  ┌─────────────────┐   │ │
│  │            │  │ Histone H1      │   │ │ Scientific section
│  │            │  │ (protein desc)  │   │ │
│  │  Protein   │  └─────────────────┘   │ │
│  │  Viewer    │  ┌─────────────────┐   │ │
│  │  (3D       │  │ Control Center  │   │ │ Toggle controls
│  │  Canvas)   │  │ [Toggles] Info  │   │ │
│  │            │  └─────────────────┘   │ │
│  │            │  ┌─────────────────┐   │ │
│  │            │  │ My Journey      │   │ │ Personal section
│  │            │  │ (what I did)    │   │ │
│  └────────────┴────────────────────────┘ │
└──────────────────────────────────────────┘
```

Desktop Split (> 768px): - **Left Panel (66%)**: `<ProteinViewer>` with interactive controls - **Right Panel (33%)**: Scrollable text content

Mobile Split (768px): - **Top Half (50%)**: `<ProteinViewer>` - **Bottom Half (50%)**: Scrollable text content

Scroll Prevention

Wheel and Touch Events:

```
<div
  className="modal-overlay"
  onWheel={(e) => e.preventDefault()}
  onTouchMove={(e) => e.preventDefault()}
/>

<div
  className="modal-content"
  onWheel={(e) => e.stopPropagation()}
  onTouchMove={(e) => e.stopPropagation()}
/>
```

How It Works: - **Overlay**: `preventDefault()` blocks all scroll events - **Content**: `stopPropagation()` allows internal scrolling, prevents bubbling to overlay - **Result**: Background scroll locked, modal content scrolls freely

---

ProteinViewer Component

Purpose: Three.js canvas that renders GLB protein models with interactive controls and hover detection.

File Details

File: `/src/components/about/ProteinViewer.jsx` (500+ lines)

Tech Stack: - **React Three Fiber**: React bindings for Three.js - **drei**: Helper components (useGLTF, OrbitControls, Html) - **Three.js**: WebGL rendering engine - **Framer Motion**: AnimatePresence for hover tooltips

Component Props

```
<ProteinViewer
  path={histoneH1}                  // GLB model path
  position={[0, 0, 0]}              // Model position [x, y, z]
  scale={[1.0, 1.0, 1.0]}           // Model scale [x, y, z]
  rotation={[0, 0, 0]}              // Initial rotation [x, y, z] (optional)
  cameraZ={40}                      // Camera Z position
  tooltip="Histone H1 (PDB: 1HST)"  // Hover tooltip text
  expanded={false}                  // Card preview (false) or modal (true)
  onHoverChange={setHoveredStructure} // Callback for hover state

  // Toggle states (controlled by parent)
  showLysines={showLysines}
  showSurface={showSurface}
  showHelices={showHelices}
  showSheets={showSheets}
  showChromophore={showChromophore}
  showInteractions={showInteractions}

  // Toggle callbacks
  onToggleLysines={() => setShowLysines(!showLysines)}
  onToggleSurface={() => setShowSurface(!showSurface)}
  onToggleHelices={() => setShowHelices(!showHelices)}
  onToggleSheets={() => setShowSheets(!showSheets)}
  onToggleChromophore={() => setShowChromophore(!showChromophore)}
  onToggleInteractions={() => setShowInteractions(!showInteractions)}
/>
```

GLB Model Loading

useGLTF Hook (from drei):

```
const { scene } = useGLTF(path);
```

How It Works: 1. **Parse GLB**: Loads binary GLTF file (geometry + materials) 2. **Return Scene**: Three.js scene graph with meshes 3. **Caching**: Automatically caches models (doesn't reload on re-render)

Model Files: - `HistoneH1_V2.glb` (13 KB) - Histone H1 linker protein - `GFP_v2.glb` (24 KB) - Green Fluorescent Protein - `BRCA1.glb` (18 KB) - BRCA1 tumor suppressor - `Cas9prot.glb` (32 KB) - Cas9 nuclease - `RNApol2.glb` (45 KB) - RNA Polymerase II

Object Naming Convention

GLB objects named by structure type:

```
Struct_Helix              → Alpha helices
Struct_Sheet              → Beta sheets
Struct_Coil               → Coils/loops
Mol_Surface               → Molecular surface
Protein                   → Full protein outline
Balls_Lysine              → Lysine residues (spheres)
Sticks_Lysine             → Lysine bonds (cylinders)
Balls_Chromophore_SYG     → GFP chromophore (spheres)
Sticks_Chromophore_SYG    → GFP chromophore (bonds)
Interaction_HBonds        → Hydrogen bonds
```

Why This Matters: - **Selective Visibility**: Toggle specific elements by name - **Color Assignment**: Apply theme colors based on name - **Hover Detection**: Identify which structure user is hovering

Color Schemes

Theme-Aware Coloring:

```
const colorSchemes = {
  dark: {
    Struct_Helix: {
      color: '#00ff88',           // Bright neon green/teal
      emissive: '#00ff88',        // Matching emissive for glow
      emissiveIntensity: 0.2
    },
    Struct_Sheet: {
      color: '#60d5ff',           // Bright neon light blue
      emissive: '#60d5ff',
      emissiveIntensity: 0.2
    },
    Struct_Coil: {
      color: '#00e5cc',           // Bright teal
      emissive: '#14b8a6',
      emissiveIntensity: 0.3
    },
    Mol_Surface: {
      color: '#ffffff',
      emissive: '#ffffff',
      emissiveIntensity: 0.1,
      opacity: 0.25,
      transparent: true
    }
  },
  light: {
    Struct_Helix: {
      color: '#ff9ec2',           // Pink
      emissive: '#ff9ec2',
      emissiveIntensity: 0.15
    },
    Struct_Sheet: {
      color: '#ffd966',           // Yellow
      emissive: '#ffd966',
      emissiveIntensity: 0.15
    },
    Struct_Coil: {
      color: '#ffb899',           // Light peach/orange
      emissive: '#ff8e53',
      emissiveIntensity: 0.2
    },
    Mol_Surface: {
      color: '#78B4DC',
      emissive: '#64C8DC',
      emissiveIntensity: 0.05,
      opacity: 0.2,
      transparent: true
    }
  }
};
```

Emissive Glow: - **Purpose**: Makes structures "glow" in dark mode - **Implementation**: `emissive` color + `emissiveIntensity` - **Effect**: Cyberpunk aesthetic, easier to see in dark theme

Visibility Control

Card vs. Modal Mode:

```
// Lysines: ONLY visible in expanded modal
if (object.name === 'Sticks_Lysine' || object.name === 'Balls_Lysine') {
  object.visible = expanded && showLysines;
}

// Surface: Always visible in card, respects toggle in modal
if (object.name === 'Mol_Surface') {
  object.visible = expanded ? showSurface : true;
}

// Helices: Always visible in card, respects toggle in modal
if (object.name === 'Struct_Helix') {
  object.visible = expanded ? showHelices : true;
}

// Sheets: Always visible in card, respects toggle in modal
if (object.name === 'Struct_Sheet') {
  object.visible = expanded ? showSheets : true;
}

// Protein overlay: Always visible (shows connecting loops)
if (object.name === 'Protein') {
  object.visible = true;
}
```

Why Different Rules? - **Card Mode**: Show full protein (no distractions) - **Modal Mode**: Allow user to deconstruct and explore

Hover Detection

Raycasting (3D mouse picking):

```
const { camera, mouse, raycaster } = useThree();

useFrame(() => {
  // Update raycaster with current mouse position
  raycaster.setFromCamera(mouse, camera);

  // Find intersected objects
  const intersects = raycaster.intersectObjects(scene.children, true);
  if (intersects.length > 0) {
    const hoveredObject = intersects[0].object;

    // Call parent's onHoverChange callback
    if (onHover) {
      onHover(hoveredObject.name);
    }

    // Increase emissive glow on hover
    if (hoveredObject.material) {
      hoveredObject.material.emissiveIntensity = 0.6; // Boost glow
    }
  }
});
```

How It Works: 1. **Raycaster**: Casts ray from camera through mouse position 2. **Intersects**: Returns array of intersected objects (sorted by distance) 3. **Hover Effect**: Increase `emissiveIntensity` for glow 4. **Parent Callback**: `onHoverChange(hoveredObject.name)` updates parent state

Result: Hovering protein → Glows + Text highlights in description

OrbitControls

```
<OrbitControls
  enableZoom={expanded}           // Only allow zoom in modal
  enablePan={expanded}            // Only allow pan in modal
  enableRotate={true}             // Always allow rotation
  minDistance={20}
  maxDistance={100}
/>
```

Why Conditional Controls? - Card Mode: Rotation only (simple interaction) - **Modal Mode**: Full control (zoom, pan, rotate)

Scientific Text Highlighting

ScientificText Component

Purpose: Renders protein descriptions with color-coded structure terms that highlight on hover.

Implementation

```
function ScientificText({ text, hoveredStructure, isDark, isMobile }) {
  const termToStructure = {
    'alpha-helix': 'Struct_Helix',
    'alpha helix': 'Struct_Helix',
    'beta-sheet': 'Struct_Sheet',
    'beta sheet': 'Struct_Sheet',
    'lysine': 'Balls_Lysine',
    'lysine residues': 'Balls_Lysine',
    'surface': 'Mol_Surface',
    'chromophore': 'Balls_Chromophore_SYG',
    // ... more mappings
  };

  const renderColoredText = () => {
    let remainingText = text;
    const parts = [];

    // Regex pattern from all terms (case insensitive)
    const pattern = new RegExp(
      `(${Object.keys(termToStructure).join('|')})`,
      'gi'
    );

    let match;
    while ((match = pattern.exec(remainingText)) !== null) {
      const term = match[0];
      const structureName = termToStructure[term.toLowerCase()];
      const isHovered = hoveredStructure === structureName;
      const baseColor = colorScheme[structureName];

      parts.push(
        <span
          style={{
            color: baseColor,
            fontWeight: isHovered ? '700' : '600',
            textShadow: isHovered
              ? `0 0 12px ${baseColor}, 0 0 6px ${baseColor}`
              : 'none',
            transition: 'all 0.2s ease'
          }}
        >
          {term}
        </span>
      );
    }

    return parts;
  };

  return <p>{renderColoredText()}</p>;
}
```

How It Works

1. **Regex Matching**: Find all structure terms in text (case-insensitive)
2. **Color Mapping**: Map term → structure name → color
3. **Hover Detection**: Check if `hoveredStructure === structureName`
4. **Dynamic Styling**: Apply color + glow if hovered

Example Text:

```
"Histone H1 is a linker histone with lysine residues that bind DNA.
The alpha-helix regions provide structural support."
```

Rendered:

```
Histone H1 is a linker histone with [lysine residues](green) that bind DNA.
The [alpha-helix](teal) regions provide structural support.
```

On Hover: - User hovers `Struct_Helix` in protein viewer - `hoveredStructure` state updates to `"Struct_Helix"` - "alpha-helix" text gets text-shadow glow + bold weight

Result: Interactive link between 3D model and text description

---

Toggle Control System

iOS-Style Toggle Buttons

Design: Animated switch buttons with gradient backgrounds.

Component Example (Surface Toggle):

```
<button
  onClick={() => setShowSurface(!showSurface)}
  style={{
    padding: 0,
    background: 'transparent',
    border: 'none',
    cursor: 'pointer',
    width: '110px',
    height: '36px'
  }}
>
  <div style={{
    position: 'relative',
    width: '100%',
    height: '100%',
    background: showSurface
      ? (isDark
          ? 'linear-gradient(135deg, #60a5fa 0%, #3b82f6 25%, #06b6d4 75%, #14b8a6 100%)'
          : 'linear-gradient(135deg, #ff6b35 0%, #ff8e53 25%, #ff6b9d 75%, #e91e63 100%)')
      : 'rgba(120, 120, 128, 0.16)',
    borderRadius: '18px',
    transition: 'background 0.3s cubic-bezier(0.4, 0.0, 0.2, 1)'
  }}>
    <motion.div
      animate={{ x: showSurface ? 76 : 4 }}
      transition={{
        type: "spring",
        stiffness: 400,
        damping: 28
      }}
      style={{
        position: 'absolute',
        width: '24px',
        height: '24px',
        background: '#ffffff',
        borderRadius: '50%',
        boxShadow: '0 2px 4px rgba(0, 0, 0, 0.2)'
      }}
    />
    <span style={{
      position: 'absolute',
      left: showSurface ? '12px' : '34px',
      fontSize: '0.9rem',
      fontWeight: showSurface ? '500' : '400',
      color: showSurface
        ? (isDark ? '#0b0b0b' : '#ffffff')
        : (isDark ? 'rgba(255, 255, 255, 0.5)' : 'rgba(0, 0, 0, 0.5)'),
      transition: 'all 0.3s cubic-bezier(0.4, 0.0, 0.2, 1)'
    }}>
      Surface
    </span>
  </div>
</button>
```

Anatomy of Toggle Button

Container (110px × 36px): - **Background**: Gradient when ON, gray when OFF - **Border Radius**: 18px (pill shape)

Sliding Circle (24px diameter): - **Position**: Animated left/right with Framer Motion - **Animation**: Spring physics (stiffness: 400, damping: 28) - **Movement**: `x: 4px` (OFF) → `x: 76px` (ON)

Label Text: - **Position**: Moves with toggle state - **Color**: Inverts based on state (dark text on light bg when ON) - **Font Weight**: Bolder when ON (500 vs 400)

Conditional Toggles

Histone H1 and Others (IDs 1, 3, 4, 5): - Surface Toggle - Lysines Toggle - Helices Toggle - Sheets Toggle

GFP (ID 2): - Ghostly Surface Toggle - Beta Barrel Toggle (controls both helices + sheets) - Chromophore Toggle - Stabilizing Bonds Toggle (hydrogen bonds)

Why Different Controls? - **GFP**: Barrel structure (combined helices + sheets) - **Others**: Separate helices and sheets

Hover Info Display

Right Side of Control Center:

```
{hoveredStructure ? (
  <div style={{
    padding: '1rem 1.25rem',
    background: `${color}20`,  // 20% opacity of structure color
    border: `2px solid ${color}`,
    borderRadius: '8px'
  }}>
    <div style={{ fontSize: '1.1rem', fontWeight: '700', color: color }}>
```

```
      {structureNames[hoveredStructure]}
    </div>
    <div style={{ fontSize: '0.9rem', color: 'rgba(255, 255, 255, 0.8)' }}>
      {structureDescriptions[hoveredStructure]}
    </div>
  </div>
) : (
  <div style={{ /* Placeholder text */ }}>
    Hover over the protein structure to see detailed information
  </div>
)}
```

How It Works: 1. **Hover Protein**: User hovers `Struct_Helix` 2. **State Updates**: `setHoveredStructure("Struct_Helix")` 3. **Info Card Appears**: Shows "Alpha Helix" + description 4. **Color Match**: Card border/background matches structure color

Result: Real-time feedback as user explores protein

Animations and Transitions

Timeline Card Animations

Desktop Entry (fade + slide):

```
<motion.div
  className="about-card right-card"
  initial={{ opacity: 0, x: 40 }}        // Start 40px right, invisible
  whileInView={{ opacity: 1, x: 0 }}     // Fade in, slide to position
  viewport={{ once: true, amount: 0.3 }} // Trigger when 30% visible
  transition={{ duration: 0.4, delay: 0, ease: "easeOut" }}
/>
```

Mobile Entry (3D card flip):

```
<motion.div
  className="about-card"
  initial={{ opacity: 0, y: -300, scale: 0.7, rotateX: -25 }}  // Above screen, tilted back
  whileInView={{ opacity: 1, y: 0, scale: 1, rotateX: 0 }}     // Drop down, face forward
  viewport={{ once: true, amount: 0.3 }}
  transition={{
    duration: 0.8,
    type: "spring",
    stiffness: 100,
    damping: 15
  }}
/>
```

Why Different? - **Desktop**: Subtle slide (professional) - **Mobile**: Dramatic flip (engaging)

Timeline Dot Animations

```
<motion.div
  className="timeline-dot"
  initial={{ opacity: 0, scale: 0, rotateZ: -180 }}  // Invisible, rotated -180°
  whileInView={{ opacity: 1, scale: 1, rotateZ: 0 }} // Fade in, spin to 0°
  viewport={{ once: true, amount: 0.8 }}              // Trigger when 80% visible
  transition={isMobile
    ? { duration: 0.6, type: "spring", stiffness: 150, damping: 12 }
    : { duration: 0.3, ease: "easeOut" }
  }
/>
```

Effect: Dots spin into view as user scrolls down timeline

Laser Connector Animations

```
<motion.div
  initial={{ height: 0 }}                 // Start collapsed
  whileInView={{ height: 534 }}           // Grow to full height
  viewport={{ once: true, amount: 0.5 }}  // Trigger when 50% visible
  transition={{ duration: 1.2, ease: "easeInOut" }}
  style={{
    position: 'absolute',
    top: '66px',
    left: 'calc(50% - 1px)',
    width: '2px',
    background: 'linear-gradient(...)',  // Orange gradient
    transformOrigin: 'top'                // Grow from top
  }}
/>
```

Effect: Vertical line “draws” from top dot to bottom dot

Timeline Intro Animation

```
<motion.div
  className="timeline-intro-block"
  initial={{ opacity: 0, y: 50 }}        // Start below, invisible
  whileInView={{ opacity: 1, y: 0 }}     // Fade in, slide up
  viewport={{ once: true, amount: 0.3 }}
  transition={{ duration: 0.8, delay: 0.3, ease: "easeOut" }}
/>
```

Purpose: Intro text animates in before cards start appearing

Mobile Responsiveness

Breakpoint System

Single Breakpoint: `768px`

```
useEffect(() => {
  const checkMobile = () => {
    const mobile = window.innerWidth <= 768;
    setIsMobile(mobile);
  };
  checkMobile();
  window.addEventListener('resize', checkMobile);
  return () => window.removeEventListener('resize', checkMobile);
}, []);
```

Layout Differences

Feature	Desktop (> 768px)	Mobile (768px)
Card Layout	Absolute positioned, alternating left/right	Stacked vertically, centered
Timeline Dots	Centered with absolute positioning	Inline before each card
Laser Connectors	Visible, animated	Hidden
Card Animations	Fade + slide	3D flip

Feature	Desktop (> 768px)	Mobile (768px)
Modal Split	66% left (protein) + 33% right (text)	50% top (protein) + 50% bottom (text)
Modal Insets	`2vw` horizontal, `2vh` vertical	`2vw` horizontal, `2vh` vertical
Control Grid	2-column toggles + hover info	1-column stacked

Mobile Timeline Intro

```
isMobile
  ? {
      position: "relative",
      left: "auto",
      top: "auto",
      width: "auto",
      margin: "0 1.5rem 3rem",
      padding: "0",
      textAlign: "left",
      zIndex: 2
    }
  : {
      position: "absolute",
      left: "var(--timeline-side-offset)",
      top: "50px",
      width: "var(--timeline-card-width)",
      zIndex: 2
    }
```

Mobile: Static block with margins (flows naturally) **Desktop**: Absolute positioned at top-left

Mobile Modal

Vertical Split:

```
style={{
  flex: isMobile ? '0 0 50%' : '2',        // 50% vs 66%
  height: isMobile ? '50%' : '100%',       // Half height vs full height
  flexDirection: isMobile ? 'column' : 'row'
}}
```

Why 50/50? - **Protein**: Needs minimum space for 3D interaction - **Text**: Needs scrollable area for long descriptions - **Balance**: 50/50 split keeps both usable

Touch Events

Stop Propagation:

```
<div
  onWheel={(e) => e.stopPropagation()}
  onTouchMove={(e) => e.stopPropagation()}
>
  {/* Modal content */}
</div>
```

Purpose: Allows modal content to scroll while preventing background scroll

---

Performance Considerations

GLB Model Caching

useGLTF Automatic Caching:

```
const { scene } = useGLTF(path);
```

How It Works: - **First Load**: Downloads GLB, parses geometry, caches in memory - **Subsequent Loads**: Returns cached scene instantly - **Memory**: All 5 models cached (~132 KB total)

Benefit: Opening same protein modal → instant render (no download)

Material Cloning

```
scene.traverse((object) => {
  if (object.isMesh && object.material) {
    object.material = object.material.clone();
  }
});
```

Why Clone? - **Problem**: Modifying shared material affects all instances - **Solution**: Clone creates unique material per object - **Cost**: ~1-2ms per clone (negligible)

Conditional Rendering

Laser Connectors:

```
{!isMobile && <motion.div ... />}
```

Why Skip on Mobile? - **Visual Clutter**: No horizontal space for connectors - **Performance**: 4 fewer animated elements (60fps → smoother) - **UX**: Vertical scroll doesn't benefit from connectors

Animation Performance

GPU-Accelerated Properties: - `opacity`: GPU compositing - `transform` (x, y, scale, rotate): GPU transform - `height`: **NOT GPU-accelerated** (but acceptable for one-time animation)

Why Height Animation? - **Laser Connectors**: Only way to animate vertical line growth - **Once Per View**: Triggered once on scroll into view - **Short Duration**: 1.2s animation completes quickly

Raycasting Optimization

useFrame Hook:

```
useFrame(() => {
  raycaster.setFromCamera(mouse, camera);
  const intersects = raycaster.intersectObjects(scene.children, true);
  // ... hover logic
});
```

Performance: - **Runs Every Frame**: 60 times per second - **Cost**: ~0.5-1ms per frame (raycasting is fast) - **Optimization**: Only checks visible objects

Future Enhancement: - Throttle raycasting to 30fps (every other frame) - Early exit if modal not open

---

Theme Integration

ThemeContext Usage

```
import { useTheme } from "../../contexts/ThemeContext";
```

```
function AcademicJourney() {
  const { isDark } = useTheme();

  // Pass to ProteinViewer
  <ProteinViewer isDark={isDark} />
}
```

Theme-Dependent Colors

Laser Connectors:

```
background: isDark
  ? 'linear-gradient(to bottom, rgba(255, 255, 255, 0.35), rgba(200, 200, 200, 0.25))'
  : 'linear-gradient(to bottom, #ffffff, #fef3cd, #fed7aa, #fb923c)'
```

Modal Background:

```
background: isDark
  ? 'rgba(0, 0, 0, 0.85)'  // Dark semi-transparent black
  : 'rgba(250, 248, 246, 0.5)'  // Light cream
```

Toggle Gradient:

```
background: showSurface
  ? (isDark
      ? 'linear-gradient(135deg, #60a5fa 0%, #3b82f6 25%, #06b6d4 75%, #14b8a6 100%)'  // Blue
      : 'linear-gradient(135deg, #ff6b35 0%, #ff8e53 25%, #ff6b9d 75%, #e91e63 100%)') // Orange/Pink
  : 'rgba(120, 120, 128, 0.16)'  // Gray
```

Protein Color Schemes

Dark Theme: Neon cyberpunk (bright greens, blues, teals) **Light Theme**: Pastel warmth (pinks, yellows, peaches)

Why Different? - Dark Mode: High contrast for visibility - **Light Mode**: Softer colors for readability

---

Future Enhancements

Potential Additions

1. **Animated Timeline Path**
   - **Current**: Straight laser connectors
   - **Enhancement**: Curved bezier paths using `TimelineSegment` component
   - **Effect**: Organic, flowing timeline
2. **Protein Annotations**
   - **Current**: Hover for structure info
   - **Enhancement**: Click on residue → Show annotation label in 3D space
   - **Implementation**: `<Html>` component from drei
3. **Timeline Filtering**
   - **Current**: All 5 experiences always visible
   - **Enhancement**: Filter by type (Education, Research, Clinical)
   - **Implementation**: Buttons above timeline toggle experience visibility
4. **Expanded Protein Library**
   - **Current**: 5 proteins (one per experience)
   - **Enhancement**: Multiple proteins per experience (user-selectable)
   - **Implementation**: Dropdown in modal to switch proteins
5. **Mobile VR Mode**
   - **Current**: OrbitControls for rotation
   - **Enhancement**: Device orientation controls (gyroscope)
   - **Implementation**: `DeviceOrientationControls` from drei
6. **Export Protein View**
   - **Current**: No sharing
   - **Enhancement**: Export current protein view as PNG
   - **Implementation**: `gl.render()` + `toDataURL()` + download
7. **Accessibility Improvements**
   - **Current**: Basic keyboard support (Escape to close)
   - **Enhancement**: Full keyboard navigation (Tab through toggles, Enter to activate)
   - **Implementation**: Focus management + ARIA labels
8. **Loading Placeholders**
   - **Current**: `<Suspense fallback={null}>`

- **Enhancement**: Skeleton loaders for protein models
- **Implementation**: Placeholder geometry while GLB loads

Related Documentation

- ABOUT-SECTION.md - Previous section (Hola + Skills Banner)
- WORK-SECTION.md *(coming soon)* - Next section (Interactive Lab)
- HERO-SECTION.md - DNA helix interaction patterns
- ARCHITECTURE.md - Component hierarchy
- STATE-MANAGEMENT.md - Theme context
- MASTER-OVERVIEW.md - Full portfolio overview

Quick Reference

Key Files

File	Lines	Purpose
`AcademicJourney.jsx`	1,469	Main timeline with modal system
`ProteinViewer.jsx`	500+	Three.js protein renderer
`TimelineSegment.jsx`	31	Curved line renderer (unused)
`ThemeContext.jsx`	50	Dark/light theme state

Timeline Positions (Desktop)

Experience	Year	Y-Position
Universidad Europea	2016	50px
University of Queensland	2019	600px
IGM Team	2022	1150px
University of Technology Sydney	2022	1400px
Lanzarote & Beyond	2025	1950px

Laser Connector Heights

Connector	From → To	Height
1	2016 → 2019	534px
2	2019 → 2022	534px
3	2022 → 2022 (IGM/UTS)	784px
4	2022 → 2025	Variable

Toggle States

Control	Default	Proteins
Surface	ON	All
Lysines	OFF	Histone H1
Helices	ON	All except GFP
Sheets	ON	All except GFP
Beta Barrel	ON	GFP only
Chromophore	ON	GFP only
Stabilizing Bonds	OFF	GFP only

Animation Durations

Element	Duration	Trigger
Timeline Card (Desktop)	0.4s	Scroll into view (30%)
Timeline Card (Mobile)	0.8s (spring)	Scroll into view (30%)
Timeline Dot	0.3s (desktop), 0.6s (mobile)	Scroll into view (80%)
Laser Connector	1.2s	Scroll into view (50%)
Timeline Intro	0.8s (delay 0.3s)	Scroll into view (30%)

This timeline transforms a traditional CV into an interactive scientific narrative, demonstrating technical skills (Three.js, WebGL) while communicating personal journey.