Hero Section Documentation

Last Updated: January 13, 2026 **Component Path:** `/src/components/hero/Hero.jsx`
Related Docs: ARCHITECTURE.md | THREE-JS-COMPONENTS.md

Table of Contents

Overview

Purpose

The Hero section is the **first impression** of the portfolio - an immersive 3D landing page featuring an interactive DNA double helix. It establishes the bio informatics theme and provides a memorable entry point.

Key Features

- **3D DNA Helix:** Animated B-form DNA structure (PDB 1BNA) with 12 base pairs
- **Auto-Animation:** Entry wiggle → continuous spin with gradual slow-down
- **Inspect Mode:** Interactive exploration with OrbitControls
- **Info Panels:** Detailed nucleotide information on hover
- **Scroll Locking:** Prevents scrolling during DNA inspection
- **Theme-Aware:** Adapts colors and effects based on dark/light mode
- **Mobile Optimized:** Vertical orientation, touch controls

User Experience Flow

```
Page loads
    ↓
DNA helix enters with wiggle animation (3 seconds)
    ↓
DNA begins spinning (fast → slow over 90 seconds)
    ↓
Title + subtitle fade in (3.2 seconds after load)
    ↓
Navbar fades in (2 seconds after load)
    ↓
"Inspect DNA" button appears (pulsing animation)
    ↓
User clicks "Inspect DNA"
    ↓
DNA stops spinning, page scrolls to top, scroll locked
    ↓
OrbitControls enabled - user can rotate/pan/zoom
    ↓
Info panels appear (DNA details + controls)
    ↓
User hovers nucleotides → nucleotide details shown
    ↓
User clicks "Click to scroll"
    ↓
DNA resumes spinning, scroll unlocked, user can explore portfolio
```

Component Structure

Component Hierarchy

```
Hero.jsx (Orchestrator)
│
├── DNAHelixScene.jsx (Three.js Canvas)
│   ├── Canvas (React Three Fiber)
│   ├── Lighting Setup
│   ├── DNA.jsx (3D Model + Interactions)
│   └── CustomOrbitControls (Conditional)
│
├── HeroContent.jsx (Title + CTA)
│   ├── motion.div (Title)
│   ├── motion.p (Subtitle)
```

```
│   └── motion.button (Inspect/Scroll Toggle)
│
└── InfoPanels.jsx (Information Display)
    ├── Desktop Layout
    │   ├── Controls Instructions (left)
    │   ├── DNA Details (right)
    │   └── Nucleotide Details (left, on hover)
    └── Mobile Layout
        ├── DNA Details (top)
        └── Nucleotide Details (overlay)
```

File Organization

Hero Component Files

```
src/components/hero/
├── Hero.jsx                # Main orchestrator - state & logic
├── DNAHelixScene.jsx       # Three.js canvas setup
├── HeroContent.jsx         # Title, subtitle, CTA button
└── InfoPanels.jsx          # Information panels for inspect mode

src/components/
└── DNA.jsx                 # 3D DNA model component (shared)

src/Styles/
└── Hero.css                # Hero section styles

src/assets/
└── Metallic_1BNA_3.0.glb   # DNA 3D model file
```

Why This Structure?

- **Hero.jsx:** Central state management for complex interactions
- **DNAHelixScene.jsx:** Isolates Three.js setup from business logic
- **HeroContent.jsx:** Separates text content for easy editing
- **InfoPanels.jsx:** Large component with detailed content, kept separate
- **DNA.jsx:** Reusable 3D component (could be used elsewhere)

State Management

Hero.jsx State Variables

```jsx
// Animation State
const [isSpinning, setIsSpinning] = useState(true);          // Spin vs inspect mode
const [animationFinished, setAnimationFinished] = useState(false);  // Entry animation done
const [wiggleDuration, setWiggleDuration] = useState(0);     // Entry wiggle duration

// UI Visibility
const [showText, setShowText] = useState(false);             // Title/subtitle visible
const [showNavbar, setShowNavbar] = useState(false);         // Navbar visible
const [showInspectButton, setShowInspectButton] = useState(false);  // CTA button visible
const [showInfo, setShowInfo] = useState(false);             // Info panels visible
const [forceInfoVisible, setForceInfoVisible] = useState(false);    // Force info in inspect mode

// Nucleotide Interaction
const [hoveredNucleotide, setHoveredNucleotide] = useState(null);  // Hovered base name

// Transition State
const [isTransitioning, setIsTransitioning] = useState(false);      // Prevent double-click
const [exitingToScroll, setExitingToScroll] = useState(false);     // Exiting inspect mode
```

State Flow Diagram

```
Initial State:
├─ isSpinning: true (spinning)
├─ animationFinished: false (not ready)
├─ showText: false (hidden)
└─ showInspectButton: false (hidden)

After 2 seconds:
└─ showNavbar: true ———→ Navbar fades in

After 3.2 seconds:
└─ showText: true ———→ Title + subtitle fade in

After wiggle animation completes (~3s):
├─ animationFinished: true
└─ showInspectButton: true ———→ "Inspect DNA" button appears

User clicks "Inspect DNA":
├─ isSpinning: false ———→ DNA stops spinning
├─ showInfo: true ———→ Info panels appear
└─ scroll locked ———→ Page can't scroll
```

```
User hovers nucleotide:
└─ hoveredNucleotide: "Adenine" ──────→ Nucleotide panel shows

User clicks "Click to scroll":
├─ isSpinning: true ──────→ DNA resumes spinning
├─ showInfo: false ──────→ Info panels hide
└─ scroll unlocked ──────→ Page can scroll
```

Animation Timeline

Entry Sequence (First 3 Seconds)

Timeline:

```
t=0s
├─ DNA model loads (Suspense)
├─ Scroll locked
└─ Entry wiggle animation begins

t=0.5s → t=3s
└─ DNA wiggles (GSAP animation from GLB)

t=2s
└─ Navbar fades in (Framer Motion)

t=3s
├─ Wiggle completes
├─ Continuous spin begins (fast)
└─ "reveal-ui" class added to body

t=3.2s
└─ Title + subtitle fade in (Framer Motion)

t=4s
└─ "Inspect DNA" button fades in (Framer Motion)

t=3s → t=93s
└─ Spin gradually slows (fast → 10% speed over 90 seconds)
```

Timing Constants

```
// /src/components/hero/Hero.jsx

const HERO_TEXT_DELAY_MS = 3200;        // Title/subtitle delay
const HERO_NAV_DELAY_MS = 2000;         // Navbar delay
const HERO_TITLE_STAGGER_SEC = 0;       // Title animation delay
const HERO_SUBTITLE_STAGGER_SEC = 0.25; // Subtitle animation delay
const INSPECT_FADE_DURATION = 1.5;      // Button fade-in duration
```

Animation Technologies

Element	Technology	Duration	Easing
DNA Wiggle	GSAP (from GLB)	3s	Custom
DNA Spin	GSAP + useFrame	90s	Exponential
Title/Subtitle	Framer Motion	1.2s	easeOut
CTA Button	Framer Motion	1.5s	easeOut
Info Panels	Framer Motion	0.4-0.6s	easeOut
Theme Toggle	CSS + Framer	0.3s	ease

DNA Helix Implementation

3D Model Details

Source: PDB 1BNA (B-form DNA) - **Structure:** 12 base-pair fragment - **Sequence:** CGCGAATTCGCG (palindromic) - **Resolution:** 1.9 Å - **Published:** 1981 (Drew et al.) - **Format:** GLB (GLTF binary) - **File:** `Metallic_1BNA_3.0.glb`

DNA.jsx Component

```
// /src/components/DNA.jsx

export default function DNA({
  isSpinning,
  setShowInfo,
  setHoveredNucleotide,
  onWiggleDuration,
  ...props
}) {
```

```
  const { scene, animations } = useGLTF(DNA_URL);
  const { isDark } = useTheme();

  // Entry animations (wiggle + spin slowdown)
  useEffect(() => {
    const wiggle = actions["DNA_WiggleAction"];
    const spin = actions["DNA_ControllerAction"];

    // Play wiggle once
    wiggle.setLoop(THREE.LoopOnce, 1);
    wiggle.timeScale = 4;  // 4x speed
    wiggle.play();

    // Continuous spin with slowdown
    spin.setLoop(THREE.LoopRepeat, Infinity);
    spin.timeScale = 4;  // Start fast
    spin.play();

    // Slow down over 90 seconds
    gsap.to(spin, {
      timeScale: 0.4,  // End at 10% of initial speed
      duration: 90,
      ease: "power2.out"
    });
  }, [actions, mixer]);

  // Raycasting for hover detection
  useFrame(() => {
    if (!isSpinning && hoverActive) {
      raycaster.setFromCamera(mouse, camera);
      const intersects = raycaster.intersectObjects(scene.children, true);

      if (intersects.length > 0) {
        const obj = intersects[0].object;
        // Determine nucleotide type from mesh name
        const baseName = determineBaseName(obj.name);
        setHoveredNucleotide(baseName);
        applyHighlight(obj, baseName);
      } else {
        clearHighlights();
      }
    }
  });

  return <primitive object={scene} {...props} />;
}
```

Nucleotide Color Scheme

Base	Color	Hex	Pairing	Bonds
Adenine (A)	Green	#40ff40	Thymine	2 H-bonds
Thymine (T)	Red	#ff4040	Adenine	2 H-bonds
Cytosine (C)	Blue	#0080ff	Guanine	3 H-bonds
Guanine (G)	Yellow	#ffff00	Cytosine	3 H-bonds
Backbone	Gray	#888888	N/A	Phosphodiester

Theme-Aware Styling

Dark Mode: - Neon emissive glow on nucleotides (40% intensity) - Blue-tinted backbone (#4A9EFF) - Strong emissive highlights on hover

Light Mode: - No emissive glow (cleaner look) - Transparent backbone (25% opacity) for text contrast - Reddish backbone hover (#E74C3C)

Inspect Mode

Entering Inspect Mode

Trigger: User clicks "Inspect DNA" button

Actions: 1. Scroll to top (if not already there) 2. Lock scroll (`prevent-scroll` class on body) 3. Stop DNA spinning (`isSpinning = false`) 4. Enable Orbit-Controls 5. Show info panels (staggered animation) 6. Enable nucleotide hover detection 7. Change button text to "Click to scroll"

Code:

```
// /src/components/hero/Hero.jsx

const toggleSpin = () => {
  if (isTransitioning) return;
  setIsTransitioning(true);

  if (isSpinning) {
    // Entering inspect mode
    const currentScrollY = window.scrollY;

    if (currentScrollY > 0) {
      // Not at top - scroll to top first
```

```
      unlockScroll();
      window.scrollTo(0, 0);

      requestAnimationFrame(() => {
        lockScroll();
        setIsSpinning(false);
        setTimeout(() => setIsTransitioning(false), 500);
      });
    } else {
      // Already at top
      lockScroll();
      setIsSpinning(false);
      setTimeout(() => setIsTransitioning(false), 500);
    }
  } else {
    // Exiting inspect mode (handled separately)
    exitInspectMode();
  }
};
```

Inspect Mode UI

Desktop:

```
┌─────────────────────────────────────────────────┐
│  [Controls Instructions]    [DNA Detail Panel] │
│  (left)                     (right)            │
│                                                │
│              [DNA 3D Model]                    │
│                                                │
│  [Nucleotide Detail]                           │
│  (left, on hover)                              │
│                                                │
│              [Click to scroll]                 │
└─────────────────────────────────────────────────┘
```

Mobile:

```
┌─────────────────────────────────────────────────┐
│  [DNA Detail Panel - Top]                      │
│                                                │
│              [DNA 3D Model]                    │
│                                                │
│  [Nucleotide Detail - Overlay on hover]        │
```

```
│                                                    │
│              [Click to scroll]                     │
└────────────────────────────────────────────────────┘
```

OrbitControls

Enabled When: `!isSpinning` (inspect mode only)

Controls: - **Left-click + drag:** Rotate camera around DNA - **Right-click + drag:** Pan camera - **Scroll wheel:** Zoom in/out - **Touch (mobile):** - 1 finger drag: Rotate - 2 finger drag: Pan - Pinch: Zoom

Implementation:

```jsx
// /src/components/hero/DNAHelixScene.jsx

function CustomOrbitControls({ isSpinning }) {
  return !isSpinning ? <OrbitControls /> : null;
}
```

Nucleotide Hover System

Raycasting:

```jsx
// /src/components/DNA.jsx

useFrame(() => {
  if (!isSpinning && hoverActive) {
    raycaster.setFromCamera(mouse, camera);
    const intersects = raycaster.intersectObjects(scene.children, true);

    if (intersects.length > 0) {
      const obj = intersects[0].object;

      // Identify nucleotide from mesh name
      if (/^A\d+/.test(obj.name)) return "Adenine";
      if (/^T\d+/.test(obj.name)) return "Thymine";
      if (/^C\d+/.test(obj.name)) return "Cytosine";
      if (/^G\d+/.test(obj.name)) return "Guanine";
      if (obj.name.includes("Backbone")) return "Backbone";

      // Update state → triggers InfoPanel
      setHoveredNucleotide(baseName);

      // Apply visual highlight
```

```
      obj.material.emissive.copy(highlightColors[baseName]);
      obj.material.emissiveIntensity = 0.5;
    }
  }
});
```

Exiting Inspect Mode

Trigger: User clicks "Click to scroll" button

Actions: 1. Hide info panels (fade out animation 0.4s) 2. Wait for panels to exit (600ms) 3. Resume DNA spinning 4. Unlock scroll 5. Change button text to "Inspect DNA"

Scroll Locking

Why Lock Scroll?

Problem: User might accidentally scroll while interacting with DNA, breaking the immersive experience.

Solution: Lock scroll during: 1. Initial entry animation (first 3 seconds) 2. Inspect mode (user exploring DNA)

Implementation

```
// /src/components/hero/Hero.jsx

const lockScroll = () => {
  const scrollY = window.scrollY;
  document.body.style.top = `-${scrollY}px`;
  document.body.classList.add("prevent-scroll");
  document.documentElement.classList.add("prevent-scroll");
};

const unlockScroll = () => {
  const scrollY = document.body.style.top;
  document.body.classList.remove("prevent-scroll");
  document.documentElement.classList.remove("prevent-scroll");
  document.body.style.top = "";

  // Restore scroll position
```

```
  if (scrollY) {
    window.scrollTo(0, parseInt(scrollY || "0", 10) * -1);
  }
};
```

CSS Implementation

```
/* /src/styles/index.css */

html.prevent-scroll,
body.prevent-scroll {
  overflow: hidden !important;
  touch-action: none !important; /* Prevent touch scrolling */
  overscroll-behavior: none !important; /* Prevent bounce */
}

body.prevent-scroll {
  position: fixed !important;
  left: 0 !important;
  right: 0 !important;
  width: 100% !important;
  height: 100vh !important;
}
```

Mobile Considerations

iOS Safari: - `position: fixed` prevents elastic scroll bounce - `touch-action: none` disables touch scroll gestures - Store and restore scroll position to prevent jump

---

Mobile Responsiveness

Detection

```
const [isMobile, setIsMobile] = useState(false);

useEffect(() => {
  const checkMobile = () => {
    setIsMobile(window.innerWidth <= 768);
  };
  checkMobile();
```

```
  window.addEventListener('resize', checkMobile);
  return () => window.removeEventListener('resize', checkMobile);
}, []);
```

DNA Orientation

Desktop: Horizontal (landscape)

```
————————ATTGC————————
```

Mobile: Vertical (portrait) - Rotated 90° clockwise

```
|
A
T
T
G
C
|
```

Implementation:

```
// /src/components/DNA.jsx

useEffect(() => {
  if (group.current) {
    if (isMobile) {
      group.current.rotation.z = Math.PI / 2;  // 90° clockwise
      group.current.scale.set(0.75, 0.75, 0.75);  // 75% scale
      group.current.position.set(0, -0.4, 0);  // Adjusted position
    } else {
      group.current.rotation.z = 0;  // Horizontal
      group.current.scale.set(1, 1, 1);  // Normal scale
      group.current.position.set(0, -0.8, 0);  // Desktop position
    }
  }
}, [isMobile]);
```

Camera Adjustments

```
// /src/components/hero/DNAHelixScene.jsx

const isMobile = typeof window !== "undefined" && window.innerWidth < 768;
const cameraPosition = isMobile ? [0, 0, -75] : [0, -2, -60];
const dnaScale = isMobile ? 1.75 : 2.65;
```

Property	Desktop	Mobile	Reason
Camera Z	-60	-75	Closer for larger DNA
DNA Scale	2.65	1.75	Smaller to fit viewport
DNA Rotation	0°	90°	Vertical for portrait
FOV	45	45	Same viewing angle

Touch Controls

Mobile Inspect Mode: - **1 finger drag:** Rotate DNA - **2 finger drag:** Pan view - **Pinch:** Zoom in/out

Touch Hints (Mobile):

```
 Drag to rotate
 Two fingers to pan
 Pinch to zoom
```

Info Panel Layout

Desktop: - Controls instructions (left) - DNA details (right) - Nucleotide details (left, replacing controls)

Mobile: - DNA details (top, centered) - Nucleotide details (overlay, centered) - Touch controls hint included in DNA panel

---

User Interactions

Button States

"Inspect DNA" Button:

```
className={`spin-toggle ${inspectReady && isSpinning ? 'heavy-pulse' : ''}`}
```

States: 1. **Hidden** (first 4 seconds) - `opacity: 0` 2. **Pulsing** (ready to inspect) - `heavy-pulse` animation 3. **Active** (inspect mode) - "Click to scroll" text

CSS:

```css
.spin-toggle.heavy-pulse {
  animation: pulse 2s ease-in-out infinite;
}

@keyframes pulse {
  0%, 100% {
    transform: translateX(-50%) scale(1);
    box-shadow: 0 0 0 0 rgba(74, 158, 255, 0.7);
  }
  50% {
    transform: translateX(-50%) scale(1.05);
    box-shadow: 0 0 0 20px rgba(74, 158, 255, 0);
  }
}
```

Theme Toggle Integration

Body Attribute:

```javascript
useEffect(() => {
  if (showInspectButton && isSpinning) {
    document.body.setAttribute('data-hero-ready', 'true');
  } else {
    document.body.removeAttribute('data-hero-ready');
  }
}, [showInspectButton, isSpinning]);
```

Theme Toggle Styling:

```css
/* Heavy pulse when hero is ready */
body[data-hero-ready="true"] .theme-toggle {
  animation: heavy-pulse 2s ease-in-out infinite;
}
```

Synced Animations: Theme toggle and Inspect DNA button pulse together

---

Performance Considerations

Optimization Techniques

1. **GLB Model Caching**

```
useGLTF.preload(DNA_URL);  // Preload on app init
```

2. **Material Cloning**

```
// Clone materials to isolate hover effects
scene.traverse((obj) => {
  if (obj.isMesh && obj.material) {
    obj.material = obj.material.clone();
  }
});
```

3. **Raycasting Optimization**

   - Only active in inspect mode (`!isSpinning && hoverActive`)
   - Throttled via `useFrame` (60fps max)
   - Single raycaster instance reused

4. **Animation Efficiency**

   - GSAP for smooth spin slowdown (GPU-accelerated)
   - Framer Motion for UI transitions (hardware-accelerated)
   - CSS transforms for button pulse

5. **Suspense Boundaries**

```
<Suspense fallback={<Html>Loading...</Html>}>
  <DNA {...props} />
</Suspense>
```

6. **Mobile Reductions**

   - 75% DNA scale (fewer pixels to render)
   - Same geometry (no model change)
   - Touch controls optimized

Performance Metrics (Target)

Metric	Desktop	Mobile	Notes
Initial Load	<2s	<3s	Until DNA visible
FPS (Spinning)	60 fps	30-60 fps	Three.js rendering
FPS (Inspect)	60 fps	30-60 fps	With OrbitControls
Memory Usage	<200MB	<150MB	Three.js + textures

Known Issues & Future Enhancements

Current Issues

1. **GFP Structure Rendering** (mentioned in known issues)
   - Some protein surfaces don't render correctly
   - Doesn't affect DNA helix
2. **Scroll Jump on iOS**
   - Occasionally scroll position jumps when exiting inspect mode
   - Mitigation: Store and restore scroll position

Potential Enhancements

1. **Prefers-Reduced-Motion**

```
const prefersReducedMotion = window.matchMedia('(prefers-reduced-motion: reduce)').matches;
if (prefersReducedMotion) {
  // Skip wiggle, use static DNA
}
```

2. **Keyboard Navigation**
   - Space/Enter to toggle inspect mode
   - Arrow keys to rotate in inspect mode
   - Escape to exit inspect mode
3. **VR/AR Support**
   - WebXR integration for VR headsets
   - AR view on mobile devices
4. **Dynamic Sequences**
   - Allow user to input custom DNA sequence
   - Generate helix on the fly
5. **Educational Tooltips**
   - Progressive disclosure of DNA facts
   - Guided tour mode

Related Documentation

- ARCHITECTURE.md - System architecture
- THREE-JS-COMPONENTS.md *(coming soon)* - 3D implementation details
- ANIMATION-SYSTEMS.md *(coming soon)* - Animation patterns
- STATE-MANAGEMENT.md - State patterns

Quick Reference

Key Props

Hero.jsx: - No props (self-contained)

DNAHelixScene.jsx: - `isSpinning`: boolean - Spin vs inspect mode - `setShowInfo`: function - Show/hide info panels - `setHoveredNucleotide`: function - Update hovered nucleotide - `onWiggleDuration`: function - Callback for wiggle duration

DNA.jsx: - `scale`: number - DNA size multiplier - `isSpinning`: boolean - Spin vs inspect mode - `setShowInfo`: function - Update info panel visibility - `setHoveredNucleotide`: function - Update hover state - `onWiggleDuration`: function - Report animation duration

HeroContent.jsx: - `showText`: boolean - Show title/subtitle - `animationFinished`: boolean - Entry animation complete - `isSpinning`: boolean - Current mode - `toggleSpin`: function - Toggle inspect mode - `disableToggle`: boolean - Prevent double-click - `inspectReady`: boolean - Show CTA button

InfoPanels.jsx: - `isSpinning`: boolean - Current mode - `showInfo`: boolean - Show info panels - `hoveredNucleotide`: string - Current hovered base - `exitingToScroll`: boolean - Exiting inspect mode

CSS Classes

- `.hero` - Main hero container
- `.hero.inspect-mode` - Hero in inspect mode
- `.dna-background` - Three.js canvas container
- `.hero-content` - Title/subtitle wrapper
- `.hero-title` - Main title
- `.hero-sub` - Subtitle
- `.spin-toggle` - CTA button
- `.spin-toggle.heavy-pulse` - Pulsing button
- `.info-panel` - Info panel container

- `.info-panel.left` - Left panel (nucleotide)
- `.info-panel.right` - Right panel (DNA details)
- `.orbit-controls-instructions` - Controls hint

This Hero section sets the tone for the entire portfolio with cutting-edge 3D graphics and thoughtful user experience.