

```

<!-- OptanonConsentNoticeStart -->
<script src="https://cdn.cookie law.org/scripttemplates/otSDKStub.js"
  type="text/javascript" charset="UTF-8"
  data-domain-script="0191e519-efb8-772f-a7bf-f57e1b4c7b18"></script>
<script type="text/javascript">
  function OptanonWrapper() { }
</script>
<!-- OptanonConsentNoticeEnd -->
<script>
  var _mtm = window._mtm = window._mtm || [];
  var _paq = window._paq = window._paq || [];
  _mtm.push({'mtm.startTime': (new Date().getTime()), 'event': 'mtm.Start'});
  var previousConsentState = localStorage.getItem('consentState') || '';

  function trackMatomo() {
    var onetrustActiveGroups = window.OptanonActiveGroups || '';
    var currentConsentState = onetrustActiveGroups.includes('C0002') ||
onetrustActiveGroups.includes('C0004');

    // Only handle tracking if the consent state has changed
    if (currentConsentState !== previousConsentState) {
      previousConsentState = currentConsentState;
      localStorage.setItem('consentState', currentConsentState);

      // Check if we just reloaded after user consent decision
      var justReloaded = localStorage.getItem('justReloaded');

      if (!justReloaded) {
        // First time user made consent decision (accept OR reject) - set flag and reload
        localStorage.setItem('justReloaded', 'true');
        location.reload();
        return;
      } else {
        // Already reloaded - remove flag and continue with appropriate tracking
        localStorage.removeItem('justReloaded');

        if (currentConsentState) {
          // Cookie-based tracking (using MTM container script)
          _paq.push(['setUserId', getUserId()]);

          (function() {
            var d = document, g = d.createElement('script'), s =
d.getElementsByTagName('script')[0];
            g.async = true;
            g.src =
'https://cdn.matomo.cloud/indegene.matomo.cloud/container_c7WRel6N.js';
            g.onload = function() {
              // Container script is loaded, enable link tracking

```

```

        // _paq.push(['enableLinkTracking']);
    };
    g.onerror = function() {
        console.error('Failed to load Matomo Tag Manager container script');
    };
    s.parentNode.insertBefore(g, s);
    })();
} else {
    // Cookie-less tracking - FIXED: Using MTM container instead of basic Matomo
    _paq.push(['disableCookies']);
    _paq.push(['setUserId', 'Anonymous--' + getAnonymousUserId()]);

    // Load MTM container script for cookie-less tracking
    (function() {
        var d = document, g = d.createElement('script'), s =
d.getElementsByTagName('script')[0];
        g.async = true;
        g.src =
'https://cdn.matomo.cloud/indegene.matomo.cloud/container_c7WRel6N.js';
        g.onload = function() {
            // Container script loaded - tags should now fire
            console.log('MTM container loaded for cookie-less tracking');
        };
        g.onerror = function() {
            console.error('Failed to load Matomo Tag Manager container script for
cookie-less tracking');
        };
        s.parentNode.insertBefore(g, s);
    })();
}
}
}
}

function getUserId() {
    let userId = localStorage.getItem('userId');
    if (!userId) {
        userId = generateUserId();
        localStorage.setItem('userId', userId);
    }
    return userId;
}

function getAnonymousUserId() {
    let anonUserId = localStorage.getItem('anonUserId');
    if (!anonUserId) {
        anonUserId = generateAnonymousUserId();
        localStorage.setItem('anonUserId', anonUserId);
    }
}

```

```
    }
    return anonUserId;
}

function generateUserId() {
  const timestamp = Date.now();
  const randomString = Math.random().toString(36).substring(2, 6);
  return `Accepted-${randomString}_${timestamp}`;
}

function generateAnonymousUserId() {
  const randomString = Math.random().toString(36).substr(2, 7);
  return `${randomString}`;
}

// Call the tracking function after Optanon has processed the consent
OptanonWrapper = function() {
  trackMatomo();
}
</script>
```