```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

 <xsl:template match="/">
  <html>
   <head>
    <title/>

<!-- xxxx  style definitions
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx -->

    <style type="text/css">
      h1 {font-family:Verdana; font-size:25px; color:#00c;}
      h3 {font-family:verdana; color:#00c}
      p {font-family:Verdana; font-size:14px; font-weight:bold;}
      span {font-family:Verdana; font-size:14px; font-weight:normal;}
      table {font-family:verdana; font-size:14px; border=0;
border-collapse:collapse;}
      th {text-align:left; border:1px solid gray; padding-left:5px;
padding-right:5px}
      th.without {text-align:left; border:0px; padding-left:5px; padding-right:5px}
      th.without_ind {color:#00c; text-align:left; border:0px; padding-left:5px;
padding-right:5px}
      th.matrix {text-align:left; vertical-align:text-top; border:1px solid gray;
padding-left:0px; padding-right:5px}
      td {text-align:left; border:1px solid gray; padding-left:5px;
padding-right:5px}
      td.without {text-align:left; border:0px; padding-left:5px; padding-right:5px}
      td.matrix {text-align:left; border:1px solid gray; padding-left:5px;
padding-right:5px;}
      td.genotype {font-family:Courier; font-size:18px; text-align:left; border:1px
solid gray; padding-left:5px; padding-right:5px}
      td.genotype2 {font-family:Courier; font-size:18px; text-align:left;
border:0px; padding-left:5px; padding-right:5px}
    </style>

   </head>

   <body>
    <xsl:apply-templates/>
   </body>
  </html>
 </xsl:template>



<!-- xxxx  header
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxx -->

 <xsl:template match="header">
    <h1>
      <xsl:value-of select="@title"/>
    </h1>

    <table>
      <tr>
        <th class="without">Number of populations:</th>
        <td class="without">
          <xsl:value-of select="number_populations"/>
        </td>
      </tr>
      <tr>
        <th class="without"> Data type: </th>
```

```
  <td class="without">
    <xsl:value-of select="data_type"/>
  </td>
</tr>

<xsl:apply-templates select="number_loci"/>

<xsl:if test="number_strains">
  <tr>
    <th class="without"> Number of strains: </th>
    <td class="without">
      <xsl:value-of select="number_strains"/>
    </td>
  </tr>
</xsl:if>

<xsl:if test="aligned">
  <tr>
    <th class="without"> Are sequences aligned: </th>
    <td class="without">
      <xsl:value-of select="aligned"/>
    </td>
  </tr>
</xsl:if>

<xsl:if test="genotypic_data">
  <tr>
    <th class="without"> Data are: </th>
    <td class="without">
      <xsl:value-of select="genotypic_data"/>
    </td>
  </tr>
</xsl:if>

<xsl:if test="missing">
  <tr>
    <th class="without"> Symbol for missing data: </th>
    <td class="without">
      <xsl:value-of select="missing"/>
    </td>
  </tr>
</xsl:if>

<xsl:if test="gap">
  <tr>
    <th class="without"> Symbol for a gap: </th>
    <td class="without">
      <xsl:value-of select="gap"/>
    </td>
  </tr>
</xsl:if>

<xsl:if test="gametic_phase">
  <tr>
    <th class="without"> Gametic phase: </th>
    <td class="without"> known </td>
  </tr>
</xsl:if>

<xsl:if test="recessive_data">
  <tr>
    <th class="without"> recessive data: </th>
    <td class="without"> yes </td>
  </tr>
</xsl:if>
```

```
    </table>
  </xsl:template>



<!-- xxxx  number loci  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

  <xsl:template match="number_loci">
    <tr>
      <th class="without"> Loci: </th>
      <td class="without"> number: </td>
      <td class="without">
        <xsl:value-of select="number" />
      </td>
    </tr>

    <xsl:for-each select="loci">
     <tr>
       <td class="without" />
       <th class="without">
         <xsl:number value="position()" format="1" />. loci:
       </th>
       <td class="without">
         <xsl:value-of select="@name" />
       </td>
     </tr>
     <tr>
       <td class="without" />
       <td class="without" />
       <td class="without"> location: </td>
       <td class="without">
         <xsl:value-of select="location" />
       </td>
     </tr>
     <tr>
       <td class="without" />
       <td class="without" />
       <td class="without"> length: </td>
       <td class="without">
         <xsl:value-of select="length" />
       </td>
     </tr>
    </xsl:for-each>

  </xsl:template>






<!-- xxxx  population
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxx -->

  <xsl:template match="population">
   <h3>
    <br />
    Population: "<xsl:value-of select="@name"/>"
   </h3>
   <p> Population size:
    <span>
     <xsl:value-of select="size" />
     <br />
```

```
    </span>

    <xsl:if test="geographic_coord">
      Geographic coordination of the population:
      <span>
        <xsl:value-of select="geographic_coord"/>
      </span>
      <br />
    </xsl:if>

    <xsl:if test="linguistic_group">
      Linguistic group of the population:
      <span>
        <xsl:value-of select="linguistic_group"/>
      </span>
      <br />
    </xsl:if>


<!-- xxxx loci begin for alignment (variable setting) xxxxxxxxxx -->

      <xsl:variable name="min">
        <xsl:for-each select="ind/strain/begin">
          <xsl:sort select="." />
          <xsl:if test="position() = 1">
            <xsl:value-of select="."/>
          </xsl:if>
        </xsl:for-each>
      </xsl:variable>

<!-- xxxxxxxxxxxxxx -->


    <xsl:if test="loci_location">
      Location of loci <xsl:value-of select="loci_location/@name" />:
      <span>
        <xsl:value-of select="loci_location" />
      </span>
      <br />
      loci begin:
      <span>
        <xsl:value-of select="$min" />
      </span>
      <br />
    </xsl:if>

    <xsl:if test="number_strains">
      Number of strains:
      <span>
        <xsl:value-of select="number_strains"/>
      </span>
      <br />
    </xsl:if>


   </p>
    <table>
      <xsl:apply-templates select="id" />
      <xsl:apply-templates select="ind">
        <xsl:with-param name="minimum" select="$min + 1" />
      </xsl:apply-templates>
    </table>
  </xsl:template>
```

```
<!-- xxxx  id
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

 <xsl:template match="id">
   <xsl:if test="position()=1">
     <tr>
       <th>identifier</th>
       <th>frequency</th>
       <xsl:if test="genotype">
         <th> genotype </th>
       </xsl:if>
       <xsl:if test="haplotype">
         <th> haplotype </th>
       </xsl:if>
     </tr>
   </xsl:if>

   <tr>
    <td> <xsl:value-of select="@name"/> </td>
    <td> <xsl:value-of select="frequency" /> </td>
    <xsl:if test="genotype|haplotype">
      <td class="genotype">
        <xsl:for-each select="genotype|haplotype">
          <xsl:value-of select="." />
          <br />
        </xsl:for-each>
      </td>
    </xsl:if>
   </tr>
 </xsl:template>




<!-- xxxx  ind
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

 <xsl:template match="ind">
  <xsl:param name="minimum" />



<!-- xxx align over one individual (different loci) xxxxxxxxxxxxxxxxxxxx-->

   <xsl:choose>
     <xsl:when test="loci_location">
       <tr>
         <th class="without_Ind"> Individual <xsl:value-of select="@name" />: </th>
       </tr>

       <xsl:if test="geographic_coord">
         <tr>
           <td class="without"> geographic coordination: </td>
           <td class="without">
             <xsl:value-of select="geographic_coord" />
           </td>
         </tr>
       </xsl:if>

       <xsl:if test="linguistic_group">
         <tr>
           <td class="without"> linguistic group: </td>
```

```
            <td class="without">
              <xsl:value-of select="linguistic_group" />
            </td>
          </tr>
        </xsl:if>

        <xsl:if test="loci_location/@name">
          <tr>
            <td class="without"> loci name: </td>
            <td class="without">
              <xsl:value-of select="loci_location/@name" />
            </td>
          </tr>
        </xsl:if>

        <xsl:if test="loci_location">
          <tr>
            <td class="without"> loci location: </td>
            <td class="without">
              <xsl:value-of select="loci_location" />
            </td>
          </tr>
        </xsl:if>


<!-- xxxx loci begin for alignment (variable setting) xxxxxxx -->

        <xsl:variable name="min">
          <xsl:for-each select="strain/begin">
            <xsl:sort select="." />
            <xsl:if test="position() = 1">
              <xsl:value-of select="."/>
            </xsl:if>
          </xsl:for-each>
        </xsl:variable>

<!-- xxxxxxxxxxxxxxxxxxx -->


        <tr>
          <td class="without"> loci begin: </td>
          <td class="without">
            <xsl:value-of select="$min" />
          </td>
        </tr>

        <xsl:if test="number_strains">
          <tr>
            <td class="without"> number of strains: </td>
            <td class="without">
              <xsl:value-of select="number_strains" />
            </td>
          </tr>
        </xsl:if>


        <tr>
          <xsl:if test="strain/genotype">
            <td class="without"> genotype: </td>
          </xsl:if>

          <xsl:if test="strain/haplotype">
            <td class="without"> haplotype: </td>
          </xsl:if>
```

```
<!-- xxxx genotype/haplotype alignment xxxxxxxxxxxxxxxx -->
          <td class="genotype2">
            <xsl:apply-templates select="strain">
              <xsl:with-param name="minimum" select="$min + 1" />
            </xsl:apply-templates>
          </td>
        </tr>

        <tr>
          <td class="without">
            <br />
          </td>
        </tr>

      </xsl:when>



<!-- xxx align over one population (ind same loci) xxxxxxxxxxxxxxxxxxxxxx-->

      <xsl:otherwise>

        <xsl:if test="position()=1">
          <tr>
            <th>individual</th>

            <xsl:if test="geographic_coord">
              <th> geographic coordination </th>
            </xsl:if>

            <xsl:if test="linguistic_group">
              <th> linguistic group </th>
            </xsl:if>

            <xsl:if test="loci_location">
              <th> loci location </th>
            </xsl:if>

            <xsl:if test="number_strains">
              <th> number of strains </th>
            </xsl:if>

            <xsl:if test="genotype">
              <th> genotype </th>
            </xsl:if>

            <xsl:if test="haplotype">
              <th> haplotype </th>
            </xsl:if>

            <xsl:if test="strain/@name">
              <th> strain name </th>
            </xsl:if>

            <xsl:if test="strain/genotype">
              <th> genotype </th>
            </xsl:if>

            <xsl:if test="strain/haplotype">
              <th> haplotype </th>
            </xsl:if>

          </tr>
```

```
        </xsl:if>

        <tr>
          <td> <xsl:value-of select="@name"/> </td>

          <xsl:if test="geographic_coord">
            <td>
              <xsl:value-of select="geographic_coord" />
            </td>
          </xsl:if>

          <xsl:if test="linguistic_group">
            <td>
              <xsl:value-of select="linguistic_group" />
           </td>
          </xsl:if>

          <xsl:if test="loci_location">
            <td>
              <xsl:value-of select="loci_location" />
            </td>
          </xsl:if>

          <xsl:if test="number_strains">
            <td>
              <xsl:value-of select="number_strains" />
            </td>
          </xsl:if>

          <xsl:if test="genotype|haplotype">
            <td class="genotype">
              <xsl:for-each select="genotype|haplotype">
                <xsl:value-of select="." />
                <br />
              </xsl:for-each>
            </td>
          </xsl:if>

          <xsl:if test="strain">
            <td class="genotype">
              <xsl:apply-templates select="strain">
                <xsl:with-param name="minimum" select="$minimum" />
              </xsl:apply-templates>
            </td>
          </xsl:if>

        </tr>

      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>



<!-- xxxx template strain xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

  <xsl:template match="strain">
   <xsl:param name="minimum" />

        <xsl:call-template name="align">
          <xsl:with-param name="min" select="$minimum" />
          <xsl:with-param name="beginning" select="begin" />

        </xsl:call-template>
        <xsl:value-of select="genotype|haplotype" />
```

```
      <br />

 </xsl:template>



<!-- xxxx template align (for alignment) xxxxxxxxxxxxxxx -->

 <xsl:template name="align">
  <xsl:param name="min" />
  <xsl:param name="beginning" />

   <xsl:if test="$beginning &gt; $min"> <xsl:call-template name="align">
       <xsl:with-param name="min" select="$min" />
       <xsl:with-param name="beginning" select="$beginning - 1" />

     </xsl:call-template>
   </xsl:if>
 </xsl:template>







<!-- xxxx  structure
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxx -->

 <xsl:template match="structure">
  <h3>
    <br />
    Structure: "<xsl:value-of select="@name"/>"
  </h3>
  <p>

    <xsl:if test="number_groups">
      Number of groups:
      <span>
        <xsl:value-of select="number_groups"/>
      </span>
      <br />
    </xsl:if>

    <xsl:if test="group">
      <xsl:for-each select="group">
        <xsl:number value="position()" format="1" />. group:
        <span>
          <xsl:value-of select="."/>
        </span>
        <br />
      </xsl:for-each>
    </xsl:if>

  </p>
 </xsl:template>
```

```
<!-- xxxx  distance matrix
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxx -->

 <xsl:template match="distance_matrix">
   <h3>
     <br />
     Distance matrix: "<xsl:value-of select="@name"/>"
   </h3>
   <p>

     <xsl:if test="size">
       Size of the distance matrix:
       <span>
         <xsl:value-of select="size"/>
       </span>
       <br />
     </xsl:if>

     Distance matrix:
   </p>

   <table>

     <td class="without" />
     <xsl:call-template name="cr2Komma2">
       <xsl:with-param name="text" select="labels" />
     </xsl:call-template>

     <xsl:call-template name="cr2br">
       <xsl:with-param name="text" select="matrix" />
       <xsl:with-param name="label" select="labels" />
     </xsl:call-template>

   </table>
   <br />
 </xsl:template>



<!-- xxxx  cr2br: seperate string after "line break"  xxxxxxxxxxxxxx -->

 <xsl:template name="cr2br">
  <xsl:param name="text" />
  <xsl:param name="label" />

  <xsl:choose>
   <xsl:when test="contains($text, '&#xA;')">
    <tr>

      <xsl:choose>
        <xsl:when test="contains($label, ',')">
          <th class="without">
            <xsl:value-of select="substring-before($label, ',')" />
          </th>
        </xsl:when>
        <xsl:otherwise>
          <th class="without">
            <xsl:value-of select="$label" />
          </th>
        </xsl:otherwise>
      </xsl:choose>


      <xsl:call-template name="cr2Komma">
```

```
        <xsl:with-param name="text" select="substring-before($text, '&#xA;')" />
      </xsl:call-template>

    </tr>

    <xsl:call-template name="cr2br">
      <xsl:with-param name="text" select="substring-after($text, '&#xA;')" />
      <xsl:with-param name="label" select="substring-after($label, ',')" />
    </xsl:call-template>

   </xsl:when>
   <xsl:otherwise>

     <tr>

       <th class="without">
         <xsl:value-of select="$label" />
       </th>

       <xsl:call-template name="cr2Komma">
         <xsl:with-param name="text" select="$text" />
       </xsl:call-template>
     </tr>

   </xsl:otherwise>
  </xsl:choose>
 </xsl:template>



<!-- xxxx  cr2br: seperate string after "comma"  xxxxxxxxxxxxxx -->

 <xsl:template name="cr2Komma">
  <xsl:param name="text" />

  <xsl:choose>
   <xsl:when test="contains($text, ',')">

     <td class="without">
       <xsl:value-of select="substring-before($text, ',')" />
     </td>

     <xsl:call-template name="cr2Komma">
       <xsl:with-param name="text" select="substring-after($text, ',')" />
     </xsl:call-template>
   </xsl:when>
   <xsl:otherwise>

     <td class="without">
       <xsl:value-of select="$text" />
     </td>

   </xsl:otherwise>
  </xsl:choose>
 </xsl:template>



<!-- xxxx  cr2br: seperate string after "comma" (for header)  xxxxxxxxxxxxxx -->

 <xsl:template name="cr2Komma2">
  <xsl:param name="text" />
  <xsl:choose>
   <xsl:when test="contains($text, ',')">
```

```
      <th class="without">
        <xsl:value-of select="substring-before($text, ',')" />
      </th>

      <xsl:call-template name="cr2Komma2">
        <xsl:with-param name="text" select="substring-after($text, ',')" />
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>

      <th class="without">
        <xsl:value-of select="$text" />
      </th>

    </xsl:otherwise>
  </xsl:choose>
 </xsl:template>



</xsl:stylesheet>
```