ldspline - Contiguous LD Representation

# Table of Contents

## Introduction

- Purpose of this manual  
- Conventions Used  
- Example commands  

```
ldspine64 report-bounds ceu dp 1.0 19 120323 120500
```

## Program Output

- Configuration details are listed first in bold- left aligned with the rest of the text.

## Common Parameters

## Using the Ldspline Application

### Summarize

```
ldspine summarize spline-filename chromosome
```

### List Splines

```
ldspine list spline-filename DP/RS float chrom bp [bp...]
```

### Reporting Boundaries

```
ldspine report-bounds spline-filename LD/RS float bp [bp][...]
```

### Expanding Boundaries

```
ldspine expand-bounds spline-filename DP/RS float chrom bp-lower bp-upper [bp-lower bp-upper][..]
```

### Load

```
ldspine load filename [filename]...
```

### Converting Splines to a Different Genome Build

```
ldspine export-lomap ldspline-filename
```

---

Ritchie Lab  

ldspline-Reference
Introduction

Purpose of this manual

Contained within this manual are details for configuring and running the application, ldspline. If this is your first time to use the software, we highly recommend that you take a few minutes to download and work through one or more tutorials. Then, once familiar with the capabilities of the software, users can refer to this guide when making changes to the basic configuration settings.

Conventions Used

There are two conventions used throughout this document. These text conventions are intended to help distinguish examples from configuration parameters.

Example commands

```
ldspline64 report-bounds ceu dp 1.0 19 120323 120500
```

The application itself will be named differently according to the platform it was compiled for. On Linux systems, it will be ldspline or ldspline64 depending on whether it runs on 32bit or 64bit systems. Windows and OS X will have be named slightly differently according to their platform as well.

Program Output

```
Program output will be in a gray box. Some lines will often be removed when the output is lengthy.
```

Configuration details are listed first in bold-left aligned with the rest of the text.

The first word(s) are the keywords which specify what is being changed. Each keyword (or phrase) has some number of parameters. These are listed in the order they should appear in the configuration line. In some cases, parameters can be repeated or are optional. Those are denoted inside []s.

Configuration details are generally followed immediately by an example line:

```
This is an example
```

Examples show how an actual entry would look and are followed by some descriptive information to help the user understand how the example would affect the ldspline application runtime.

Common Parameters
There are a number of parameters which are used commonly across multiple configuration settings. In order to simplify the descriptions of the various properties of each command, we’ll describe those properties here, and just refer to them as if they were a type.

**Integer**
Parameters specified in this way just simply refer to a whole number. In general, these values should be equal to or greater than 0, except when specified otherwise.

**Float**
Values specified as float are decimal values.

**Index**
If a parameter is listed as an index, it refers to the index, starting at 1 the user wishes to select.

**max**
This is generally an integer value representing the upper bound of some value. In some cases, such as minor allele frequency, it might represent a floating point value.

**min**
This is generally an integer value representing the lower bound of some value. In some cases, such as minor allele frequency, it is possible that it represents a floating point value.

**On/Off**
These parameters accept a boolean, Yes/No type setting. Users can use ON/OFF or YES/NO to set them.

**filename**
When a configuration refers to a file for input or output, the filename is generally used. This can be either a fully qualified path (such as /home/torstees/wga) or it can specified as a path relative to the directory where the application was run (such as ../data/goodfilename). It can also be just a plain filename as long as the file itself is available from the directory in which the application was run.

**label**
A label refers to a parameter whose value can be any text string without whitespace. These labels are generally used for reporting but in many cases are used to determine filenames. As a result, users should avoid using unusual characters in the string that could possibly cause problems with filenames. Because spaces and tabs are used to separate each parameter on a given line, labels can not contain spaces.

**description**
A description is a chunk of text that can contain spaces. It will always be at the very end of a line and is generally optional.
Using the ldspline Application

Unlike most of the applications developed by the Ritchie lab, ldspline does not use a configuration file. Instead, all variations on program execution are done at the command line. While many of these commands take similar sets of parameters, some do vary, which will be covered in the following document.

In order to more clearly describe the various parameters required, this document will present each function independently, and the appropriate parameters that are used.

In general, the first parameter is the command (summarize, list, report-bounds, etc) and the second is the spline filename. This filename can be a fully qualified filename with or without the extension .ldspline. This extension should not be changed and is assumed.

The term spline below refers to the stretch of pairwise LD both up and downstream from a single SNP where the LD remains above the given threshold. Splines can represent either R^2 squared or DPrime.

**Summarize**
The summarize report is mostly used for debugging, but it can be helpful to users to identify exactly which SNPs are found inside a given Spline file. Summary reports are based on a single chromosome.

**ldspline summarize spline-filename chromosome**

```
ldspline summarize ceu X
```

This would generate a report for the chromosome X based on the spline file ceu.ldspline that exists in the user’s current directory. Below is a small portion of the output you might see.

<table>
<thead>
<tr>
<th>chrom</th>
<th>rs</th>
<th>primary pos</th>
<th>secondary rs</th>
<th>secondary pos</th>
<th>DP</th>
<th>dir</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>rs5961354</td>
<td>5675878</td>
<td>9994064572</td>
<td>192</td>
<td>157</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>rs5961355</td>
<td>5675932</td>
<td>9994067372</td>
<td>193</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>rs5961356</td>
<td>5676132</td>
<td>9994070172</td>
<td>194</td>
<td>155</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>rs1032040</td>
<td>5677733</td>
<td>9994072972</td>
<td>194</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>rs1482826</td>
<td>5679969</td>
<td>9994075780</td>
<td>190</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>rs5961828</td>
<td>5680521</td>
<td>9994078556</td>
<td>191</td>
<td>155</td>
<td></td>
</tr>
</tbody>
</table>

### List Splines

This will generate a comprehensive list of all splines for each SNP listed. SNPs are listed by base-pair location.

**ldspline list spline-filename DP/RS float chrom bp [bp...]**

```
ldspline list ceu-b37 DP 0.8 3 122287289
```

This will generate a listing of a single spline associated with the locus 3:122287289 with a threshold of DPrime of 0.8.

In this example, we are using build 37 for the CEU population.

<table>
<thead>
<tr>
<th>chrom</th>
<th>primary rs</th>
<th>primary pos</th>
<th>secondary rs</th>
<th>secondary pos</th>
<th>DP</th>
<th>dir</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>rs9839782</td>
<td>122287289</td>
<td>rs9917729</td>
<td>122285701</td>
<td>0.8070</td>
<td>down</td>
</tr>
<tr>
<td>3</td>
<td>rs9839782</td>
<td>122287289</td>
<td>rs12486285</td>
<td>122286160</td>
<td>1.0000</td>
<td>down</td>
</tr>
<tr>
<td>3</td>
<td>rs9839782</td>
<td>122287289</td>
<td>rs2332285</td>
<td>122288210</td>
<td>0.8010</td>
<td>up</td>
</tr>
<tr>
<td>3</td>
<td>rs7629285</td>
<td>122265739</td>
<td>rs13070816</td>
<td>122234028</td>
<td>1.0000</td>
<td>down</td>
</tr>
<tr>
<td>3</td>
<td>rs7629285</td>
<td>122265739</td>
<td>rs4485669</td>
<td>122234615</td>
<td>1.0000</td>
<td>down</td>
</tr>
<tr>
<td>3</td>
<td>rs7629285</td>
<td>122265739</td>
<td>rs4677957</td>
<td>122257767</td>
<td>1.0000</td>
<td>down</td>
</tr>
<tr>
<td>3</td>
<td>rs7629285</td>
<td>122265739</td>
<td>rs28365795</td>
<td>122259640</td>
<td>1.0000</td>
<td>down</td>
</tr>
<tr>
<td>3</td>
<td>rs7629285</td>
<td>122265739</td>
<td>rs6780181</td>
<td>122264619</td>
<td>1.0000</td>
<td>down</td>
</tr>
<tr>
<td>3</td>
<td>rs7629285</td>
<td>122265739</td>
<td>rs17267388</td>
<td>122267546</td>
<td>1.0000</td>
<td>up</td>
</tr>
<tr>
<td>3</td>
<td>rs7629285</td>
<td>122265739</td>
<td>rs9289196</td>
<td>122268506</td>
<td>1.0000</td>
<td>up</td>
</tr>
</tbody>
</table>

(truncated)
Reporting Boundaries

Users can query for upper and lower bounds for splines, similar to the list function except the application will report only the upper and lower bounds.

`ldspline report-bounds spline-filename LD/RS float bp [bp][...]`

`ldspline report-bounds ceu-b37 DP 0.8 3 122287289 122265739`

Executing this command will return the boundaries for two SNPs.

Expanding Boundaries

Users can query for boundary expansion using splines. The returned boundaries is the upper and lower bound for all SNPs (splines) found within the bounded region. This allows users to quickly identify the maximum region covered by LD for a given bounded region.

`ldspline expand-bounds spline-filename DP/RS float chrom bp-lower bp-upper [bp-lower bp-upper][...]`

`ldspline expand-bounds ceu-b37 DP 0.8 3 122265738 122287238 14512345 14515123`

<table>
<thead>
<tr>
<th>chrom</th>
<th>init_lower</th>
<th>init_upper</th>
<th>lower_bound</th>
<th>upper_bound</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>122265738</td>
<td>122287238</td>
<td>122122133</td>
<td>122290579</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>14512345</td>
<td>14515123</td>
<td>14510761</td>
<td>14525027</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Load

```bash
[torstees@queso data] ~/bin/ldspline64 expand-bounds ceu-b37 DP 0.8 3 122265738 122287238 14512345 14515123
```

<table>
<thead>
<tr>
<th>chrom</th>
<th>rs</th>
<th>pos</th>
<th>lower_pos</th>
<th>lower_rs</th>
<th>upper_bound</th>
<th>upper_rs</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>rs9839782</td>
<td>122287289</td>
<td>rs9841897</td>
<td>122282569</td>
<td>rs2332285</td>
<td>122288210</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>rs7629285</td>
<td>122265739</td>
<td>rs13070816</td>
<td>122234028</td>
<td>rs2332285</td>
<td>122288210</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Converting hapmap ld data into a binary spline is possible using the LOAD command. Users should download the Hapmap data and have it extracted. A “configuration” file is then created with two columns, the first naming the chromosome and the second indicating the file containing the ld data. Below is an example of the portion of our chb configuration:

<table>
<thead>
<tr>
<th></th>
<th>filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ld_chr1_CHB.txt</td>
</tr>
<tr>
<td>2</td>
<td>ld_chr2_CHB.txt</td>
</tr>
<tr>
<td>3</td>
<td>ld_chr3_CHB.txt</td>
</tr>
<tr>
<td>4</td>
<td>ld_chr4_CHB.txt</td>
</tr>
<tr>
<td>5</td>
<td>ld_chr5_CHB.txt</td>
</tr>
<tr>
<td>6</td>
<td>ld_chr6_CHB.txt</td>
</tr>
<tr>
<td>7</td>
<td>ld_chr7_CHB.txt</td>
</tr>
<tr>
<td>8</td>
<td>ld_chr8_CHB.txt</td>
</tr>
<tr>
<td>9</td>
<td>ld_chr9_CHB.txt</td>
</tr>
<tr>
<td>10</td>
<td>ld_chr10_CHB.txt</td>
</tr>
</tbody>
</table>

(from the file, chb)

**ldspline load filename [filename]...**

*ldspline load chb*

This would parse the file, chb, converting the contents into an LD Spline binary file.

**Converting Splines to a Different Genome Build**

Users can import new base location data into splines using files generated by liftOver ([http://hgdownload.cse.ucsc.edu/admin/exe/](http://hgdownload.cse.ucsc.edu/admin/exe/)) In order to update a spline to a new build, users must perform 3 tasks:

**ldspline export-lomap ldspline-filename**

*ldspline export-lomap ceu*

This will generate a bim file compatible with the liftOver application. This will be named the same as the population with the extension of .bim added.

Users should run liftOver and specify the following parameters:

- population.new - as the new file
- population.unmapped - as the unMapped files

Where population is the name of the spline without the extension (i.e. ceu or ceu-b36)

Once the two new files have been generated, importing requires one more command:
ldspline import-lomap ldspline-filename new-build-number
  ldspline import-lomap ceu 37

The previous example will import the new position information from the files, ceu.new and ceu.unmapped and name the new spline file, ceu-b37.ldspline.