Point cloud local ancestry inference (PCLAI) Manual v0.1

Margarita Geleta,* *Alexander G. Ioannidis*

This manual provides practical guidance for users of the **Point cloud local ancestry inference (PCLAI)** outputs distributed with HPRC Release 2. It describes the structure of the released **BED9** files, including how window identifiers encode sample and haplotype information, how to interpret the **confidence score** and the **coordinate-derived color** fields. This document focuses on data usage and interpretation rather than model training details.

When using the PCLAI method or PCLAI outputs, please cite:

```
@article{geleta_pclai_2026,
    author = {Geleta, Margarita and Mas Montserrat, Daniel and
    ↪  Ioannidis, Nilah M. and Ioannidis, Alexander G.},
    title = {Point cloud local ancestry inference:
    ↪  coordinate-based ancestry along the genome},
    year = {2026}
}
```

Contents

1 Overview **2**

2 BED output format **2**

2.1 BED fields 3

2.2 Index tables and coordinate systems 4

2.3 Visualizing PCLAI outputs in IGV 4

3 PCLAI impainting for missing windows **5**

4 PCLAI discretization into labels **6**

5 FAQ **7**

* For questions, bug reports, or requests for alternative coordinate spaces/reference models, please contact the authors: geleta@berkeley.edu

1 Overview

PCLAI represents each individual as a *point cloud* along the genome, where each point corresponds to a fixed genomic window (segment) on a specific haplotype. For each window, the method outputs:

1. A continuous coordinate (by default a 2D principal component coordinate on the PC1-PC2 plane),
2. A model uncertainty/confidence summary for that coordinate.

We distribute these outputs as **BED** file (Section 2), with per-window colors encoding the inferred coordinate and a numeric score encoding confidence (Figure 1).

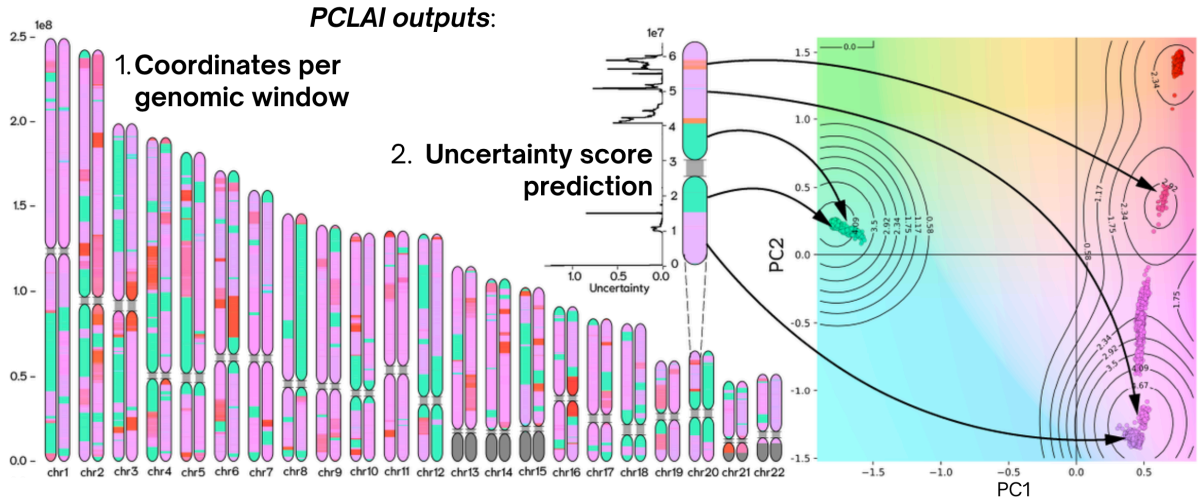

Figure 1: **PCLAI output overview.** PCLAI infers a continuous coordinate for each fixed genomic window along a haplotype (left; "chromosome painting" where each window is colored by its inferred position in the coordinate space) and an accompanying uncertainty/confidence score for each window (center; summarized per window, reported as the BED `score` in the range 0–1000). Coordinates are inferred by regressing each window into a reference 2D PCA space (right; shown on the PC1–PC2 plane), enabling visualization of local ancestry as a continuous trajectory across the genome. Output is distributed as a BED9 track in which `itemRgb` encodes the coordinate-derived color and the `score` encodes confidence.

Data sources and preprocessing Local ancestry painting in this release was performed using variant calls derived from Minigraph-Cactus (MC). Briefly, the MC VCF was converted to a **biallelic** representation and imputed with Minimac4 to harmonize the input with the PCLAI training representation. PCLAI was then run on autosomes to infer a continuous coordinate for each genomic window, where each segement corresponds to a fixed window of **1000 SNPs**.

2 BED output format

The BED (Browser Extensible Data) file contains 9 tab-separated columns, following the UCSC Genome Browser specification[1] (Table 1). Below is an example BED9 line (GRCh38-coordinate output):

```
chr1   14486   805864   HG00097/h1/chr1_w0001_(0.43,-1.39)   991   .   14486 805864   222,162,255
```

[1] For more information on BED format, refer to `https://genome.ucsc.edu/FAQ/FAQformat.html`.

2.1 BED fields

The `name` field stores the sample, haplotype, chromosome-window identifier, and the inferred 2D coordinate. The format is:

`SAMPLE/HAP/CHROM_wXXXX_(x,y)`

where:

- `SAMPLE` is the sample ID (e.g., `CHM13`),
- `HAP` is `h0` for CHM13 and typically `h1`/`h2` for diploid samples. Most diploid assemblies contain two haplotypes which we denote `hap1` and `hap2`. The **CHM13** assembly is effectively homozygous; therefore, we treat the retained haplotype as `hap0`,
- `CHROM_wXXXX` identifies the chromosome and window index (zero-padded),
- `(x,y)` are the inferred coordinates for that window (by default, **PC1** and **PC2**).

These coordinates are directly comparable across samples *only when* they were produced using the same reference coordinate system and model. You can find the coordinates of the PCA reference in `https://github.com/AI-sandbox/hprc-pclai`.

Table 1: **BED9 fields in the distributed PCLAI output.** Definition of each column in the BED9 track produced by PCLAI.

Column	Name	Description (PCLAI usage)
1	`chrom`	Chromosome name (e.g., `chr1`, `chr22`).
2	`chromStart`	0-based start coordinate of the window (inclusive).
3	`chromEnd`	0-based end coordinate of the window (exclusive).
4	`name`	Window identifier including sample, haplotype, window index, and inferred coordinate.
5	`score`	Confidence score in $[0, 1000]$.
6	`strand`	Set to ".".
7	`thickStart`	Set to `chromStart`. Included for BED9 compatibility.
8	`thickEnd`	Set to `chromEnd`. Included for BED9 compatibility.
9	`itemRgb`	Comma-separated `R,G,B` color encoding the inferred coordinate.

The `score` column ranges from **0 (lowest confidence)** to **1000 (highest confidence)**. Conceptually, it summarizes how certain or uncertain the model is about the inferred coordinate for a window (higher score means the model is more confident in that window's coordinate prediction).

To improve interpretability and to remove extremely uncertain predictions, we **filtered out** all windows with `score < 141` prior to distribution. Therefore:

- Windows with low confidence do *not* appear in the BED file,
- Gaps in coverage can reflect filtered windows (in addition to regions without calls).

Because the score is monotonic with confidence, you can:

- Restrict analyses to high-confidence calls (e.g., `score` $\geqslant$ `threshold`),

- Compare windows near transitions versus within long homogeneous tracts.

The `itemRgb` field stores a comma-separated `R,G,B` triplet (each in $[0, 255]$). This color is a visual encoding of the inferred 2D coordinate (e.g., PC1–PC2 position) and is intended for direct genome browser visualization.

We generate colors using a **CIELAB-based** mapping across the 2D plane to improve perceptual uniformity, and then **convert to RGB** for storage in the BED file. Thus, RGB values are the *export format*, while the underlying interpolation is performed in CIELAB.

2.2 Index tables and coordinate systems

PCLAI results are distributed as BED tracks in three coordinates systems. To make it easy to locate the appropriate file for a given sample and coordinate system, we provide index CSVs in the HPRC annotation tables repository (`https://github.com/human-pangenomics/hprc_intermediate_assembly/tree/main/data_tables/annotation/pclai`):

- **PCLAI in GRCh38 coordinates:** `pclai_v0.1_grch38_coord_local_hprc_r2_v1.0.index.csv`
- **PCLAI in CHM13 coordinates:** `pclai_v0.1_chm13_coord_local_hprc_r2_v1.0.index.csv`
- **PCLAI in assembly-native coordinates:** `pclai_v0.1_asm_coord_local_hprc_r2_v1.0.index.csv`

All model training and primary inference were performed in **GRCh38 coordinates**. BED tracks in **CHM13** and **assembly-native** coordinates were generated by lifting over the GRCh38-coordinate outputs using chain files derived from the Minigraph-Cactus Release 2 alignments. Because liftOver is not guaranteed to succeed everywhere (e.g., across some centromeric regions), coordinate-transformed BEDs may exhibit additional dropouts relative to the GRCh38-coordinate tracks.

2.3 Visualizing PCLAI outputs in IGV

You can load the distributed PCLAI BED file directly in **IGV** (`File` $\rightarrow$ `Load from File`). Once loaded, IGV will display each genomic window as an interval along the chromosome (Figure 2).

The BED track provides two complementary visual summaries:

- **Coordinate-derived color:** each interval is colored using the BED `itemRgb` field, which encodes the inferred per-window coordinate (e.g., PC1–PC2 position). This enables rapid visual inspection of coordinate shifts along the genome as changes in interval color.
- **Confidence score:** the BED `score` field stores the per-window confidence score (0 = highest uncertainty; 1000 = highest confidence). When loaded as a separate track (or displayed as a value track, depending on IGV settings), this allows users to identify regions where coordinate predictions are less reliable and may have been filtered or should be interpreted cautiously.

In practice, we recommend viewing the colored BED intervals alongside the uncertainty track to distinguish genuine coordinate transitions from regions with elevated uncertainty.

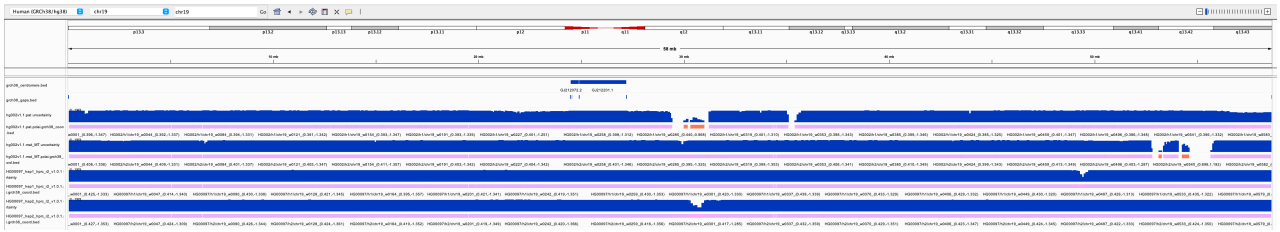

Figure 2: **Example IGV visualization of PCLAI BED outputs.** Screenshot from IGV (GRCh38/hg38) showing PCLAI tracks on chromosome 19. For each haplotype track, BED intervals are colored by `itemRgb`, which encodes the per-window inferred coordinate (e.g., PC1–PC2 position), enabling rapid visual detection of coordinate shifts along the genome. The accompanying blue value tracks display the per-window `score` (confidence; 0 = highest uncertainty, 1000 = highest confidence). Assembly gap/centromere annotations are shown above, and missing intervals can reflect filtered high-uncertainty windows and/or reference gaps.

3 PCLAI impainting for missing windows

Regions without PCLAI ancestry assignment can arise for three main reasons:

- **No source calls in the input VCFs:** difficult regions (e.g., acrocentric p-arms and other challenging loci) may lack sufficient variant calls in the training VCFs, leading to regional dropout.
- **Low-confidence filtering:** windows with low confidence scores (as assigned by PCLAI) are filtered out (windows with `score` < `141`) in the distributed BEDs (Section 2), which can appear as gaps between retained windows.
- **LiftOver dropouts (for CHM13/assembly-native tracks):** training and inference were performed in GRCh38 coordinates; CHM13- and assembly-native BEDs were produced via liftOver using Minigraph-Cactus Release 2 chains. Not all regions lift over successfully (e.g., centromeres), which can introduce additional gaps in coordinate-transformed tracks.

To decide whether it is appropriate to inpaint across a gap, we compare the inferred coordinates of the two windows that straddle the gap (the last retained window before the gap and the first retained window after the gap). We compute the L2 distance in PCA space,

$$\mathrm{d_{L2}} = \sqrt{(\Delta \mathrm{PC1})^2 + (\Delta \mathrm{PC2})^2}.$$

We then define a single global discontinuity threshold $\mathrm{x} = 0.105$ as the 0.99 quantile of $\mathrm{d_{L2}}$ computed over *contiguous* adjacent windows (i.e., neighboring windows with no genomic gap between them), pooled across all samples, haplotypes, and chromosomes. If the straddling distance exceeds this conservative threshold ($\mathrm{d_{L2}} > \mathrm{x}$), we **do not recommend inpainting**, since the straddling windows likely reflect different ancestry regimes and smooth interpolation would be misleading. If $\mathrm{d_{L2}} \leqslant \mathrm{x}$, inpainting with an average is typically reasonable for downstream summaries.

4 PCLAI discretization into labels

Although PCLAI is designed to be continuous, you can post-process inferred coordinates (e.g., with kNN boundaries in the reference space) to obtain discrete ancestry labels when required for comparison or downstream workflows.

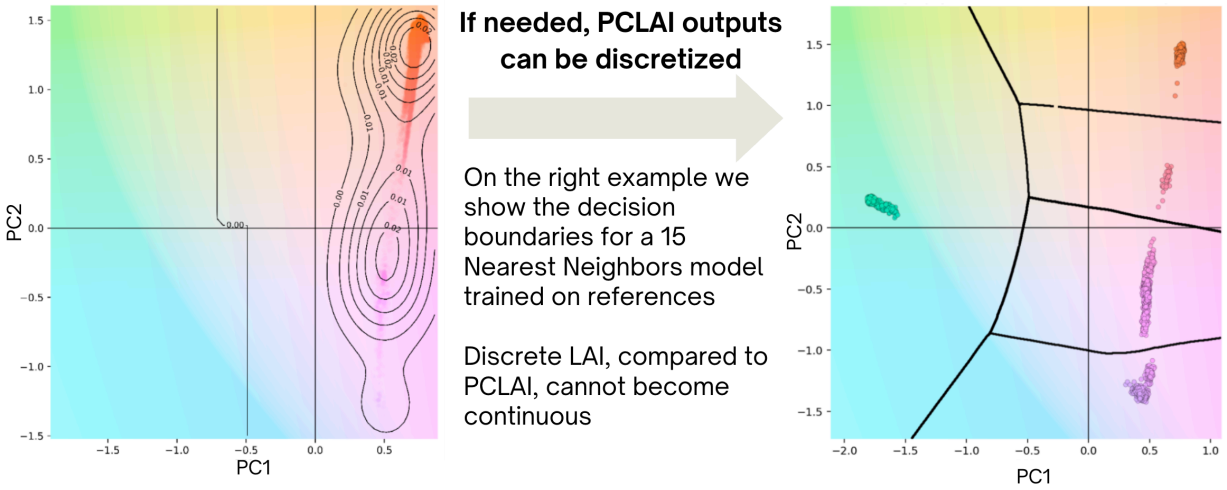

Figure 3: **Optional discretization of continuous PCLAI coordinates.** PCLAI produces continuous per-window coordinates in a reference PCA space (PC1–PC2; left), enabling ancestry variation to be represented as a continuous trajectory rather than fixed categories. When discrete labels are required for comparison or downstream workflows, coordinates can be post-processed using a classifier trained on reference samples. The right panel illustrates decision regions from a 15-nearest-neighbors (kNN) model in the same PCA space, which partitions the plane into discrete regions and maps each inferred window coordinate to a categorical assignment. This discretization is an optional overlay on the continuous PCLAI output.

When discrete labels are required, we recommend the following post-processing approach (Figure 3):

1. **Train a kNN classifier in the reference space.** Fit a **15-nearest-neighbors (kNN)** classifier using **L2 (Euclidean) distance** on the same reference PCA space used by PCLAI (you can find it in `https://github.com/AI-sandbox/hprc-pclai`), with training labels given by the **1000 Genomes five continental superpopulations** (AFR, AMR, EAS, EUR, SAS).
2. **Assign each inferred window to a superpopulation.** For each haplotype window in a sample, take the PCLAI-inferred coordinate and apply the trained kNN model to obtain a discrete superpopulation label (i.e., the region/decision boundary shown in Figure 3).
3. **Compute per-sample proportions (and an argmax label).** Aggregate window-level labels across the genome to compute the fraction of windows assigned to each superpopulation. If desired, define a single genome-wide label by the **dominant (argmax)** proportion.

Why this is recommended. In a benchmark test set of 100 samples, this discretized PCLAI summary was highly concordant with **AD-**

MIXTURE at **K=5** when comparing dominant (argmax) assignments. The confusion matrix showed 100% concordance for AFR, EAS, EUR, and SAS, with a single discordant case in which one ADMIXTURE-AMR sample was assigned AFR by discretized PCLAI. Notably, in this benchmark ADMIXTURE at K=5 did not cleanly recover an AMR cluster (PUR and MXL clustered with EUR) and AFR samples split approximately 60/40 across two clusters. Despite these behaviors, discretized PCLAI recapitulated the same broad continental structure and matched ADMIXTURE's dominant labels almost perfectly. For this reason, we recommend kNN discretization as an *optional* post-processing step for users who require categorical labels, while retaining the underlying continuous coordinates for analyses where fine-scale variation matters.

5 FAQ

Why did you filter windows using `score < 141`? In this release, the BED `score` is a *confidence score*: **`score`=0 indicates maximum uncertainty** and **`score`=1000 indicates maximum confidence**. We therefore filter to remove windows where the model is close to maximally uncertain.

Specifically, we keep windows with **`score` < 141**, so that the distributed BED track includes only regions where the model is not operating at the highest uncertainty regime. In a quick empirical check plotting genomic windows versus uncertainty (with points colored by the inferred PCA-derived color), the most uncertain windows cluster at the top of the uncertainty axis and correspond to a visually distinct group; applying the `score` threshold removes this maximally-uncertain set while preserving the remainder of the signal.

This filtering choice is also consistent with how we detect and handle discontinuities in the inferred coordinate trajectory. We compute a genome-wide breakpoint signal by measuring how much the inferred PCA coordinates jump between consecutive windows:

$$\mathrm{d}_{\mathrm{L2}} = \sqrt{(\Delta \mathrm{PC1})^2 + (\Delta \mathrm{PC2})^2}.$$

Across all chromosomes, we set a global threshold at the **0.99 quantile** of d_{L2} so that only the top 1% of jumps are treated as strong discontinuities. We apply the same principle whenever the distance between two windows straddling a gap exceeds the genome-wide 0.99 quantile, indicating the flanking windows likely come from different ancestral origins and the gap should not be smoothly impainted (Section 3).

Is the `strand` field meaningful? No. It is included for BED9 compatibility; PCLAI windows represent haplotypic segments independent of transcriptional strand.