AirLift
A Fast and Comprehensive Technique for Remapping Alignments between Reference Genomes

Jeremie S. Kim*, Can Firtina*, Meryem Banu Cavlak, Damla Senol Cali, Nastaran Hajinazar, Mohammed Alser, Can Alkan, and Onur Mutlu

bioRxiv Preprint
Source Code

SAFARI
ETH Zürich
Carnegie Mellon
SFU
Simon Fraser University
Bilkent University
Genome Analysis

- Genome analysis is critical for many applications
  - Personalized medicine
  - Outbreak tracing
  - Evolutionary studies

- Genome sequencing machines extract smaller fragments of the original DNA sequence, known as reads
Reference Genomes

- **Reference genomes** play a crucial role in genome analysis for:
  - Accurately mapping **reads** to potential **matching locations** in the genome
  - Identifying **genomic differences** in an individual’s genome

- **Reference genomes** should provide an **accurate and complete** representation of a species to **enable accurate analysis** in the **later steps of genome analysis**:
  - Variant calling
  - Gene annotation and enrichment
Updating the Reference Genomes

- Reference genomes are updated **regularly** to
  - Correct the errors in the older versions
  - Fill in the missing genomic sequences

Old Reference Genome

```
...GCCCATATCCAAAGCTTC??????AATGGGCTTAAAGCTTCCACAATG...
GCCCAAATGTTT
GCTTCCAGAATG
```

New (updated) Reference Genome

```
...GCCCATATGGTTAAGCTTCCATGGAATGGGCTTTCGCCCTTCCACAATG...
```

**Unmapped Reads**

• Remapping the reads to the updated reference genome can generate **novel information** due to
  - More **accurately** identified genomic differences
  - **New reads mapped** to updated or completed regions
Changes between Reference Genomes

1. **Retired Regions**
   - **Removed** from the new reference genome

2. **New Regions**
   - **Added** to the new reference genome

3. **Constant Regions**
   - Exactly the **same sequences**
   - Positions may change

4. **Updated Regions**
   - Mostly the same sequences with **small changes**
Existing Solutions for Remapping Reads

1. Map all the reads from scratch

2. Move the mapping locations
Existing Solutions for Remapping Reads

1. Map all the reads from scratch

2. Move the mapping locations
AirLift must preprocess look up tables between the two references in the 8 steps enumerated.

Fig. 3.

We compare two reference genomes with large sequences (i.e., regions). We identify four types of regions:

1. Constant regions
2. Updated regions
3. Extract seeds from regions
4. Map seeds from retired to constant regions

In order for AirLift to map any number of reads from an old reference genome to a new reference genome, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes to be aligned exactly (non-blue regions). Note that these seeds are the same in both old and new reference genomes, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes.

We next describe how we identify and use these regions to quickly and comprehensively remap a read set.

Fig. 3.

We compare two reference genomes with large sequences (i.e., regions). We identify four types of regions:

1. Constant regions
2. Updated regions
3. Extract seeds from regions
4. Map seeds from retired to constant regions

In order for AirLift to map any number of reads from an old reference genome to a new reference genome, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes to be aligned exactly (non-blue regions). Note that these seeds are the same in both old and new reference genomes, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes.

We next describe how we identify and use these regions to quickly and comprehensively remap a read set.

Fig. 3.

We compare two reference genomes with large sequences (i.e., regions). We identify four types of regions:

1. Constant regions
2. Updated regions
3. Extract seeds from regions
4. Map seeds from retired to constant regions

In order for AirLift to map any number of reads from an old reference genome to a new reference genome, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes to be aligned exactly (non-blue regions). Note that these seeds are the same in both old and new reference genomes, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes.

We next describe how we identify and use these regions to quickly and comprehensively remap a read set.

Fig. 3.

We compare two reference genomes with large sequences (i.e., regions). We identify four types of regions:

1. Constant regions
2. Updated regions
3. Extract seeds from regions
4. Map seeds from retired to constant regions

In order for AirLift to map any number of reads from an old reference genome to a new reference genome, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes to be aligned exactly (non-blue regions). Note that these seeds are the same in both old and new reference genomes, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes.

We next describe how we identify and use these regions to quickly and comprehensively remap a read set.

Fig. 3.

We compare two reference genomes with large sequences (i.e., regions). We identify four types of regions:

1. Constant regions
2. Updated regions
3. Extract seeds from regions
4. Map seeds from retired to constant regions

In order for AirLift to map any number of reads from an old reference genome to a new reference genome, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes to be aligned exactly (non-blue regions). Note that these seeds are the same in both old and new reference genomes, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes.

We next describe how we identify and use these regions to quickly and comprehensively remap a read set.

Fig. 3.

We compare two reference genomes with large sequences (i.e., regions). We identify four types of regions:

1. Constant regions
2. Updated regions
3. Extract seeds from regions
4. Map seeds from retired to constant regions

In order for AirLift to map any number of reads from an old reference genome to a new reference genome, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes to be aligned exactly (non-blue regions). Note that these seeds are the same in both old and new reference genomes, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two genomes.

We next describe how we identify and use these regions to quickly and comprehensively remap a read set.

Fig. 3.
Mapping Reads from Scratch

A large portion of the reference genome remains unchanged (constant regions)

Identifying the differences for reads in the constant regions is redundant

Entire Reference Genome

12% Changed Content
Existing Solutions for Remapping Reads

1. Map all the reads from scratch

2. Move the mapping locations
Existing Solutions for Remapping Reads

1. Map all the reads from scratch

2. Move the mapping locations
AirLift must preprocess look up tables between the two references in the 8 steps enumerated. Note that this alignment can be done with NX regions of approximate similarity across the reference genomes. We refer to regions that match perfectly across the old and new reference genomes, we must first either acquire an available chain file [green] from retired regions (blue). Next, we align exactly (non-blue regions). Note that these seeds are completely overlapping sequences and starting 1 base pairs later (providing 1 base pairs before each region, a seed begins at each location and ends to initially define regions that do not map to the old reference). From retired regions, we create our own. We create our chain file by running global alignment without errors between the two reference genomes, we next describe how we identify and use these regions to quickly and comprehensively remap a read set. We use exact alignment between two references marked with black bars).
Moving the Mapping Locations

- **Cannot Remap:** Reads in the **deleted regions** are not remapped

- **Bad Remap:** Reads in the **updated regions** may map other regions better

A large portion of the **mapping information is lost or inaccurate**
Accurately and quickly remap **all reads** by either **mapping or moving** them from the **old reference genome** to the **new reference genome**
Avoids redundant read mapping for the constant regions

Quickly **identifies and maps the reads** that cannot be accurately moved
AirLift must preprocess look up tables between the two references in the 8 steps enumerated. This chain file shows where exact sequences from the old reference genome can be found in the new reference that do not map to any region in the old reference genomes. This chain file shows where exact sequences from the old reference genome can be found in the new reference that do not map to any region in the old reference genomes. Note that these seeds align exactly (non-blue regions). Note that these seeds align exactly (non-blue regions). Note that these seeds align exactly (non-blue regions). Note that these seeds align exactly (non-blue regions).

We next describe how we identify and use these regions to quickly and comprehensively remap a read set. In order for AirLift to map any number of reads from an old reference genome to a new reference genome, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two reference genomes. We compare two reference genomes with large sequences (i.e., regions). We identify four types of regions: exact global alignment regions, regions of approximate similarity, constant regions, and retired regions.

We propose to generate lookup tables (LUTs) to aid in the efficient mapping of reads from one reference genome to another. Figure 3 shows the methodology for creating the LUTs. Starting with the old and new reference genomes, we must first either acquire an available chain file or generate our own. We create our chain file by running global alignment without errors between the two reference genomes. This chain file shows where exact sequences from the old reference genome can be found in the new reference that do not map to any region in the old reference genomes. This chain file shows where exact sequences from the old reference genome can be found in the new reference that do not map to any region in the old reference genomes.

Low computation overhead
Accurate Mapping
AirLift Indexing (Offline)

1. Find exactly matching regions via global alignment
   - Old Reference
   - New Reference
   - 100% match

2. Extract seeds from old reference regions that do not align exactly
   - Overlapping seeds

3. Align extracted seeds from the old reference to the new reference
   - No matches

4. Use alignment scores to initially label regions
   - Seeds from a retired region do not map to the new reference
   - Seeds from old reference do not map to a new region

5. Extract seeds from new regions (in the new reference)
   - Overlapping seeds

6. Align seeds from new regions to constant regions in old reference
   - Categorize regions that seeds align to, as updated regions

7. Form constant regions LUT based on all final constant region labels

8. Form updated regions LUT based on all final updated region labels

Constant Region  Updated Region  Retired Region  New Region
AirLift Indexing (Offline)

1. Find exactly matching regions via global alignment
2. Extract seeds from old reference regions that do not align exactly
3. Align extracted seeds from the old reference to the new reference
4. Use alignment scores to initially label regions
   - Seeds from a retired region do not map to the new reference
   - Seeds from old reference do not map to a new region
5. Extract seeds from new regions (in the new reference)
6. Align seeds from new regions to constant regions in old reference
7. Form constant regions LUT based on all final constant region labels
8. Form updated regions LUT based on all final updated region labels

Constant Region  Updated Region  Retired Region  New Region

SAFARI
AirLift must preprocess look up tables between the two references in the 8 steps enumerated.

1. Extract seeds (i.e., smaller subsequences) from regions (shown in Figure 2) that fully describe the relationship between two reference genomes:
   - Constant region
   - Updated region
   - Retired region
   - New region

2. Map seeds from retired regions to constant regions.
3. Get constant regions.
4. Get updated regions.
5. Check alignments (e.g., (1) or (2)) or
6. Determine which regions do not map to a new region.
7. Overlapping seeds
8. Unmapped reads

Quickly move reads in the constant regions
Remap reads in the updated regions
Remap retired and unmapped reads

SAFARI
AirLift Remapping

AirLift fully utilizes all reads by either moving or remapping them

AirLift generates an accurate alignment file (BAM) that can easily be used in downstream analysis
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
</tr>
<tr>
<td>Goal and Key Idea</td>
</tr>
<tr>
<td>AirLift</td>
</tr>
<tr>
<td>Evaluation</td>
</tr>
<tr>
<td>Conclusions</td>
</tr>
</tbody>
</table>
Evaluation Methodology

Remapping

• **Baseline:** Fully mapping all reads
  - CrossMap remapper that can generate alignment files (BAM)
  - LiftOver remapper that generates only the updated positions

**Accuracy:** Variant calling using AirLift and full mapping

Datasets

• **Human (hg):** Oldest: HG16 Newest: HG38 (5 versions)
• **Worm (ce):** Oldest: ce2 Newest: ce11 (5 versions)
• **Yeast (sacCer):** Oldest: sacCer1 Newest: sacCer3 (3 versions)
Performance

2.6× – 6.7× speedup compared to the full mapping

More comprehensive mapping:

Longer execution times than CrossMap and LiftOver
Peak Memory Usage

Peak memory usage similar to full mapping
# Accuracy – Variant Calling

**Precision/Recall** values compared to
- Ground truth
- Full mapping

<table>
<thead>
<tr>
<th>Remap Technique</th>
<th>Read Sets from</th>
<th>to</th>
<th>vs. Full Mapping SNP (%)</th>
<th>Indel (%)</th>
<th>vs. Ground Truth SNP (%)</th>
<th>Indel (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline:</strong></td>
<td>Full Mapping</td>
<td>-</td>
<td>hg38</td>
<td>-</td>
<td>99.54/88.00</td>
<td>81.31/92.38</td>
</tr>
</tbody>
</table>

Comparable accuracy to full mapping without the significant performance cost
# AirLift Summary

### Problem
Remapping to a new reference genome is either **costly (full mapping)** or **inaccurate (moving mapping positions)**

### Goal
Accurately and quickly remap **all reads** by either mapping or moving them from the **old reference genome** to the **new reference genome**

### AirLift
- **AirLift Indexing:** Accurately categorize and label each region in the old reference genome compared to the new reference genome
- **AirLift Remapping:**
  1. Remap a read to a new reference genome or
  2. Quickly move its position based on **AirLift index**

### Key Results
AirLift **consistently outperforms full mapping**
- **2.6x – 6.7x speedup** over full mapping

AirLift identifies SNPs and INDELs with precision and recall **similar to full mapping**
AirLift

• Jeremie S. Kim, Can Firtina, Meryem Banu Cavlak, Damla Senol Cali, Nastaran Hajinazar, Mohammed Alser, Can Alkan, and Onur Mutlu,
"AirLift: A Fast and Comprehensive Technique for Remapping Alignments between Reference Genomes"
[bioRxiv preprint]
arXiv preprint
[AirLift Source Code and Data]
AirLift Source Code

AirLift is a tool that updates mapped reads from one reference genome to another. Unlike existing tools, it accounts for regions not shared between the two reference genomes and enables remapping across all parts of the references. Described by Kim et al. (preliminary version at http://arxiv.org/abs/1912.08735)

https://github.com/CMU-SAIFARI/AirLift
AirLift
A Fast and Comprehensive Technique for Remapping Alignments between Reference Genomes

Jeremie S. Kim*, Can Firtina*, Meryem Banu Cavlak, Damla Senol Cali, Nastaran Hajinazar, Mohammed Alser, Can Alkan, and Onur Mutlu

bioRxiv Preprint
Source Code
P&S Genomics

Lecture 12b: AirLift

Can Firtina

ETH Zürich
Spring 2023
26 May 2023
Backup Slides
AirLift Remapping

1. **Read data set & mapping information to old reference (BAM file)**

   For each read that mapped to old reference
   
   1. **Check mapping location to old reference in constant regions LUT**
      
      If read mapped to a constant region
      
      1. Remap the read using any remapping tool (e.g., CrossMap)
   
   If read did not map to any constant region
   
   2. **Check mapping location to old reference in updated regions LUT**
      
      If read mapped to an updated region
      
      2. Remap the read to the new reference using a full mapper (e.g., BWA-MEM)
   
      If read did not map to any updated region
      
      3. **The read mapped to a retired region in the old reference**
         
      3. Mark read as unmapped in the new reference
   
   4. Remap the read to new and updated regions in the new reference using a full mapper (e.g., BWA-MEM)