Fast Algorithms for Improved Transcriptome Analysis I : Transcriptomic Mapping

Rob Patro

CSE 549 Fall 2016

COMputational **BI**ology and **N**etwork **E**volution

website: https://combine-lab.github.io/

We're interested in a wide range of comp. bio problems:

- Biological network evolution
- Chromatin structure & epigenetic regulation
- Data representation & storage:
  - *Dynamic* text indexing
  - short-read compression
- Computational transcriptomics
  - Efficient read mapping
  - Transcript-level expression inference
  - transcriptome assembly & analysis

COMputational **BI**ology and **N**etwork **E**volution

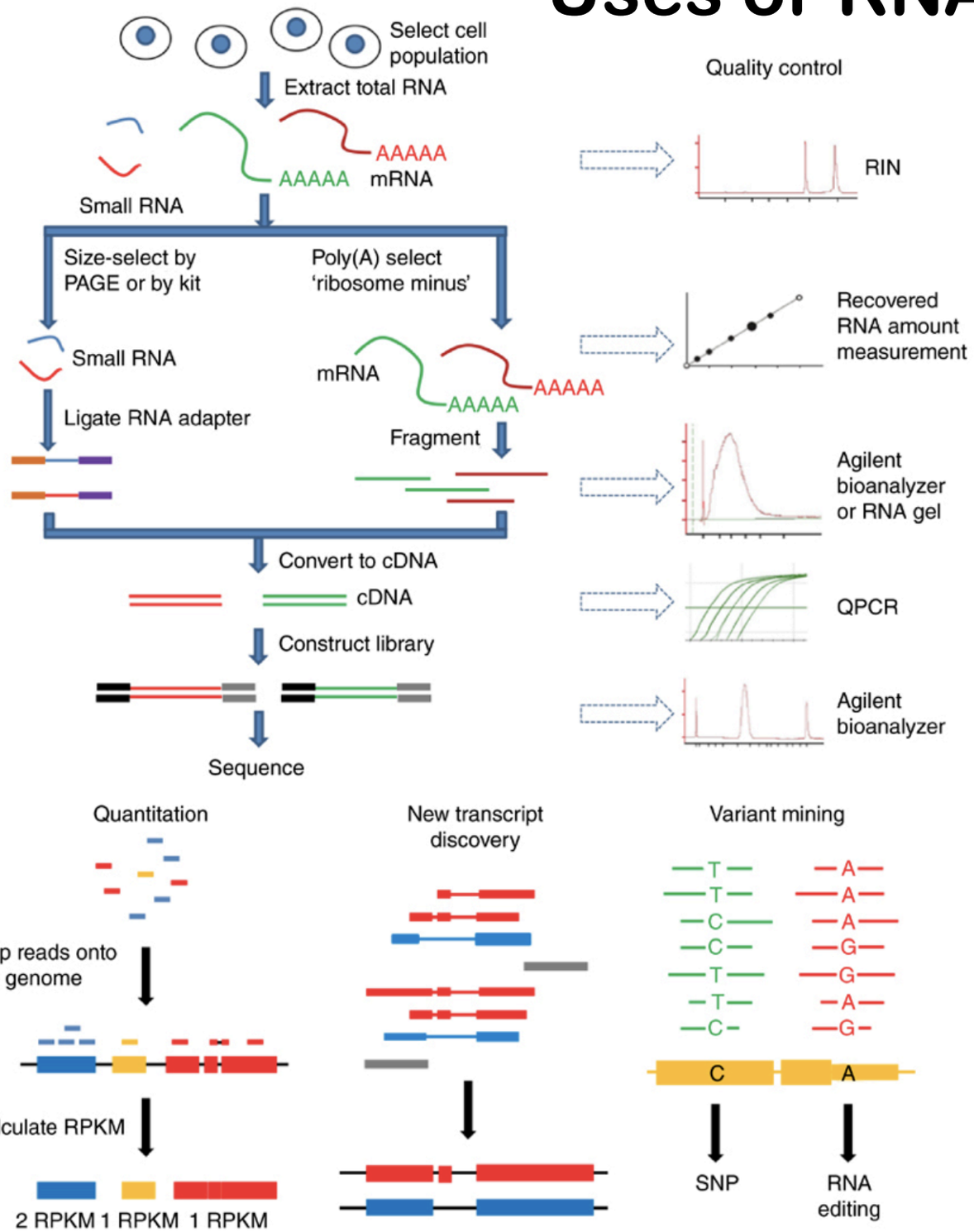
website: https://combine-lab.github.io/

We're interested in a wide range of comp. bio problems:

- Biological network evolution
- Chromatin structure & epigenetic regulation
- Data representation & storage:
  - *Dynamic* text indexing
  - short-read compression
- Computational transcriptomics (this and the next lecture)
  - Efficient read mapping
  - Transcript-level expression inference
  - transcriptome assembly & analysis

Zeng & Mortazavi, Nature Immunology 2012

Uses of RNA-Seq are manifold



Whole transcriptome analysis

- Quantification & differential expression
- Novel txp discovery
  - reference-based
  - *de novo*
- Variant detection
  - Genomic SNPs
  - RNA editing

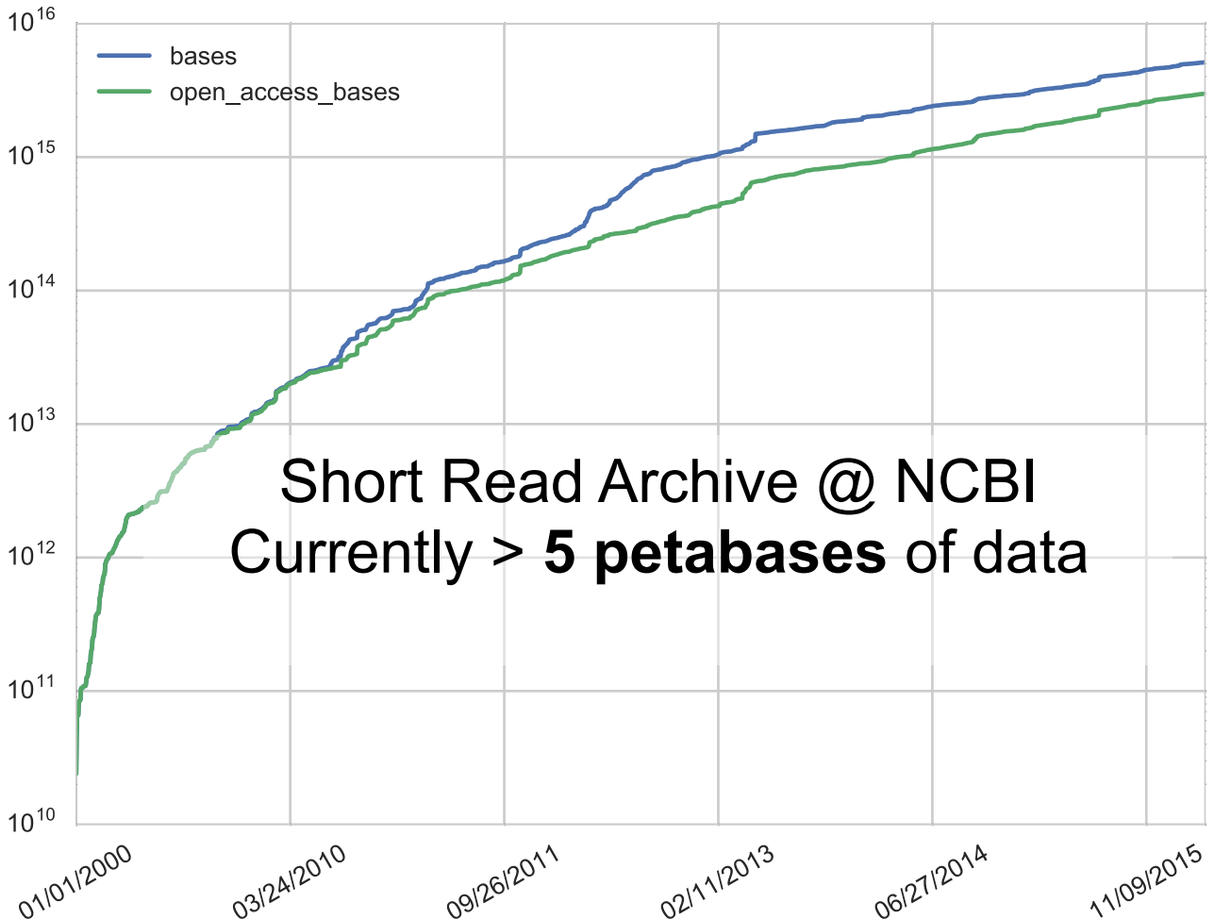
- What is dynamic & changing over time (as disease progresses)?
- What is tissue specific (in fetal development but not after)?
- What is condition specific (under stress conditions vs. not)?

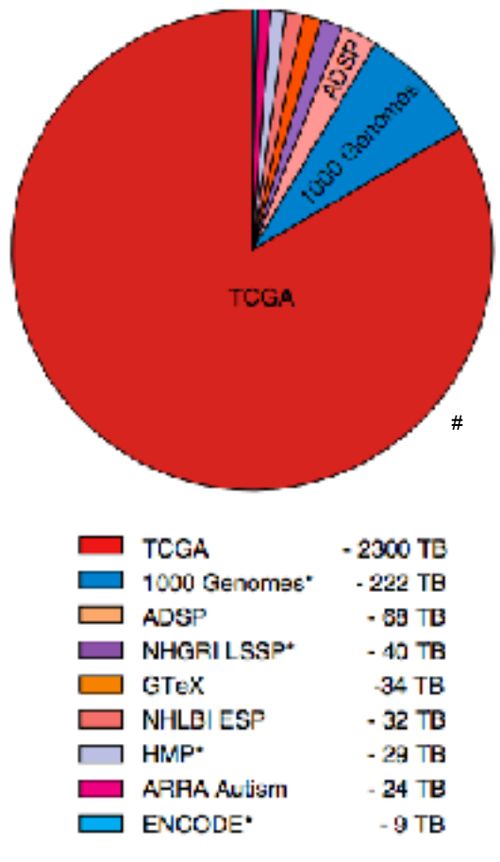
Why do we still need faster analysis?

OPINION Open Access

The real cost of sequencing: scaling computation to keep pace with data generation

Paul Muir[1,2,3], Shantao Li[4], Shaoke Lou[4,5], Daifeng Wang[4,5], Daniel J Spakowicz[4,5], Leonidas Salichos[4,5], Jing Zhang[4,5], George M. Weinstock[6], Farren Isaacs[1,2], Joel Rozowsky[4,5] and Mark Gerstein[4,5,7*]





In addition to new data, re-analysis of existing experiments often desired: In light of new annotations, discoveries, and methodological advancements.

[#] From Fig 1 of Muir et al.

Advocating for analysis-efficient computing

- Compute *only* the information required for your analysis; ask what information you *need* to solve your problem, not what output current tools are generating
- Often the efficiency of the analysis is related to the *size* of the (processed) data's representation
- Not all analyses require such efficient solutions, should concentrate on problems where this is actually needed.

I'll provide some (hopefully) compelling examples:

- **RapMap**: Read alignment → quasi-mapping (get "core" info much faster)
- **Salmon**: Fast, state-of-the-art quantification using quasi-mapping, dual-phase inference & fragment eq. classes
- **RapClust**: Fast, accurate *de novo* assembly clustering using quasi-mapping & fragment eq. classes

We believe these ideas are **general**, and can be applied to many problems

Advocating for analysis-efficient computing

- Compute *only* the information required for your analysis; ask what information you *need* to solve your problem, not what output current tools are generating
- Often the efficiency of the analysis is related to the *size* of the (processed) data's representation
- Not all analyses require such efficient solutions, should concentrate on problems where this is actually needed.

I'll provide some (hopefully) compelling examples:

Boiler (by your very own Pritt & Langmead) is also a beautiful example of this idea.

When we have a particular analysis in mind — transcript identification & quantification — we can compress data much more aggressively & effectively.

We believe these ideas are **general**, and can be applied to many problems

Sequencing Reads

pre-proc.

align to ref.

txp. identification

reference-based

pre-proc.

de novo assembly

align to assembly

de novo

quantification

DE, Alternative Splicing, etc.

“Higher-level” analysis

RNA-Seq Read Alignment

Given an RNA-seq read, where *might* it come from?

Two main “regimes”

Align to transcriptome

Align reads directly to txps

No “split” alignments — transcripts contain spliced exons directly.

Typically *a lot* of multi-mapping (80-90% of reads may map to multiple places)

Does *not r*equire target *genome*

Can be used in *de novo* context (i.e. after *de novo* assembly)

Align to genome

Align reads to target genome

Reads spanning exons will be “split” (gaps up to 10s of kb)

Typically little multi-mapping (most reads have single genomic locus of origin)

Requires target *genome*

Can be used to find new transcripts

RNA-Seq Read Alignment

Given an RNA-seq read, where does it come from?

Two main “regimes”

Align to transcriptome

Main computational challenge comes from ubiquitous multi-mapping.

Bowtie

Bowtie 2

BWA

STAR

HISAT (1&2)

…

Align to genome

Main computational challenge comes from spliced alignments.

Top Hat

STAR

HISAT (1&2)

Map Splice

Subread Aligner

…

RNA-Seq Read Alignment

Given an RNA-seq read, where does it come from?

Two main “regimes”

Align to transcriptome

Main computational challenge comes from ubiquitous multi-mapping.

We’ll focus on this “regime” today.

Bowtie

Bowtie 2

BWA

STAR

Align to genome

Main computational challenge comes from spliced alignments.

Top Hat

STAR

HISAT (1&2)

Map Splice

Subread Aligner

Problem 1: RNA-Seq Read ~~Alignment~~ Mapping

What if we don't *need* alignment?

Claim: Some (but not all) of the analyses we're interested in performing may ***not actually require the read alignment***

How much more efficient may a solution be if we only care about ***where*** and not exactly ***how*** a read corresponds to the reference?

Validation: For a *very common analysis*, RNA-seq-based quantification and differential expression testing, we can replace alignment with mapping with virtually **no loss in accuracy.**

RNA-Seq Read Alignment

Alignment is ***fast*** . . . but not always as fast as our data is ***big***

A single ***sample*** may contain 10s of millions of reads

An ***experiment*** may consist of many samples
e.g. conditions, time course samples, etc.

Condition A	Condition B	Condition C	Condition D	Condition E
Replicate 1	Replicate 1	Replicate 1	Replicate 1	Replicate 1
Replicate 2	Replicate 2	Replicate 2	Replicate 2	Replicate 2
Replicate 3	Replicate 3	Replicate 3	Replicate 3	Replicate 3
Replicate 4	Replicate 4	Replicate 4	Replicate 4	Replicate 4

A single *experiment* may easily consist of **100s of millions of reads.**

Quasi-mapping: A stand-in for alignment

Concept:

For a given fragment, a quasi-mapping specifies the ***target*** where a fragment "matches well", and the ***position***, and ***orientation*** of the fragment w.r.t the target, but *not details of the alignment*.

Algorithm:

Relies on a suffix array to compute the *Maximum Mappable Prefix* (MMP) and *Next Informative Position* (NIP) when mapping a read.

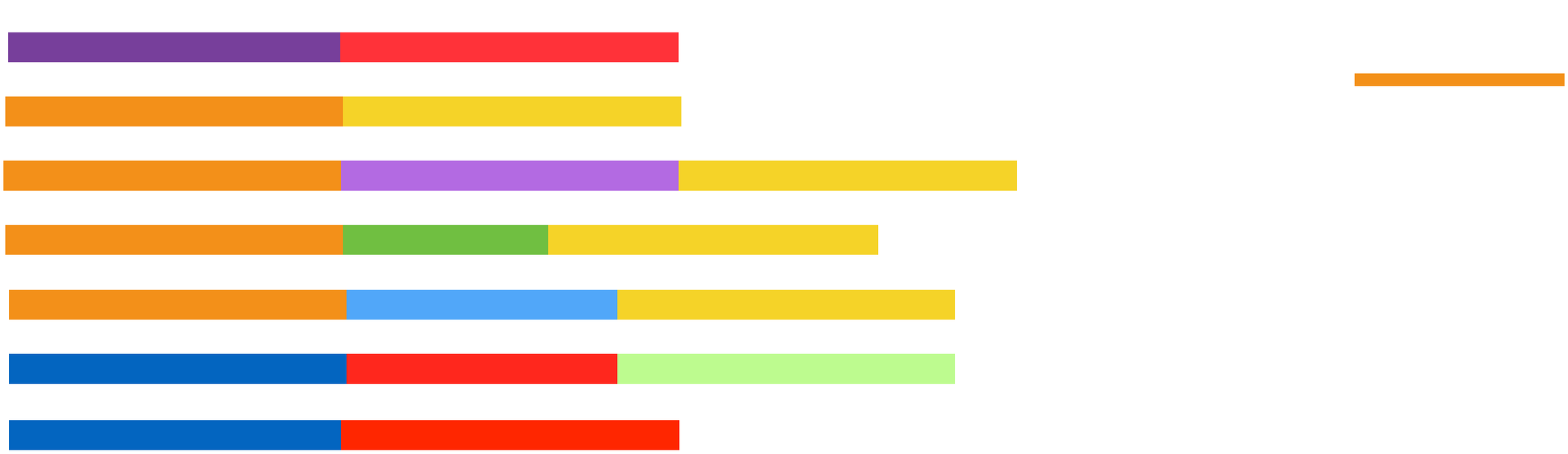
Given a carefully-designed algorithm, quasi-mapping information can be obtained *very* quickly.

Mapping reads to a Transcriptome

Consider the following scenario:

Transcripts

Read



Mapping reads to a Transcriptome

Consider the following scenario:

Say that colors represent exonic sequence.
Intuitively, from where does the read originate?

Transcripts

Read

Mapping reads to a Transcriptome

Consider the following scenario:

Say that colors represent exonic sequence.
Intuitively, from where does the read originate?
What about *this* read?

Transcripts

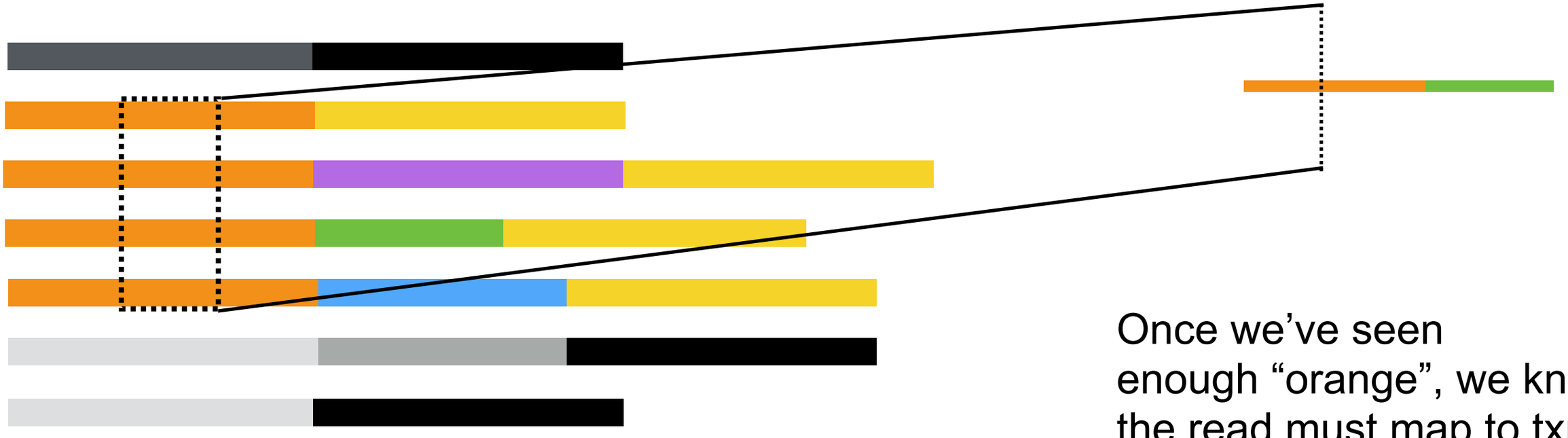
Read

Mapping reads to a Transcriptome

Consider the following scenario:

Transcripts

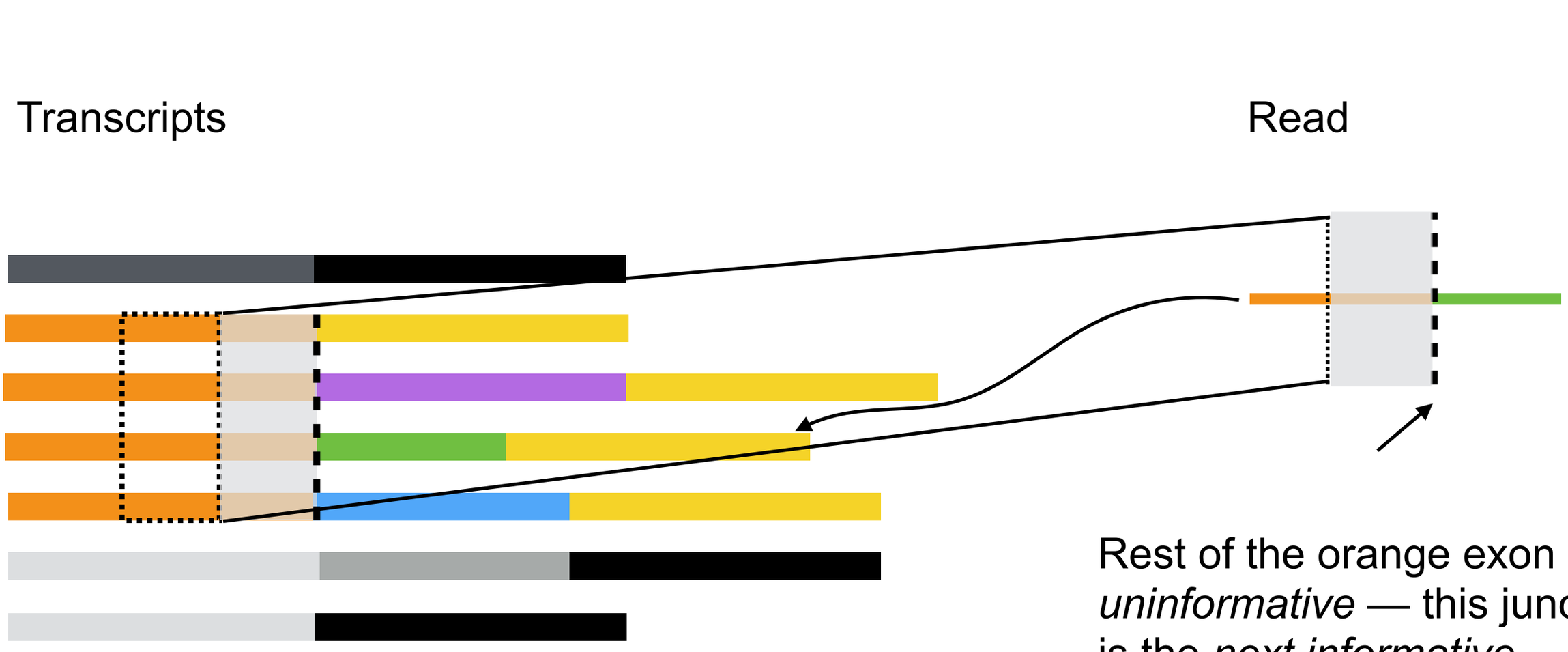
Read



Once we've seen enough "orange", we know the read must map to txps with this exon; but which one(s)?

Mapping reads to a Transcriptome

Consider the following scenario:

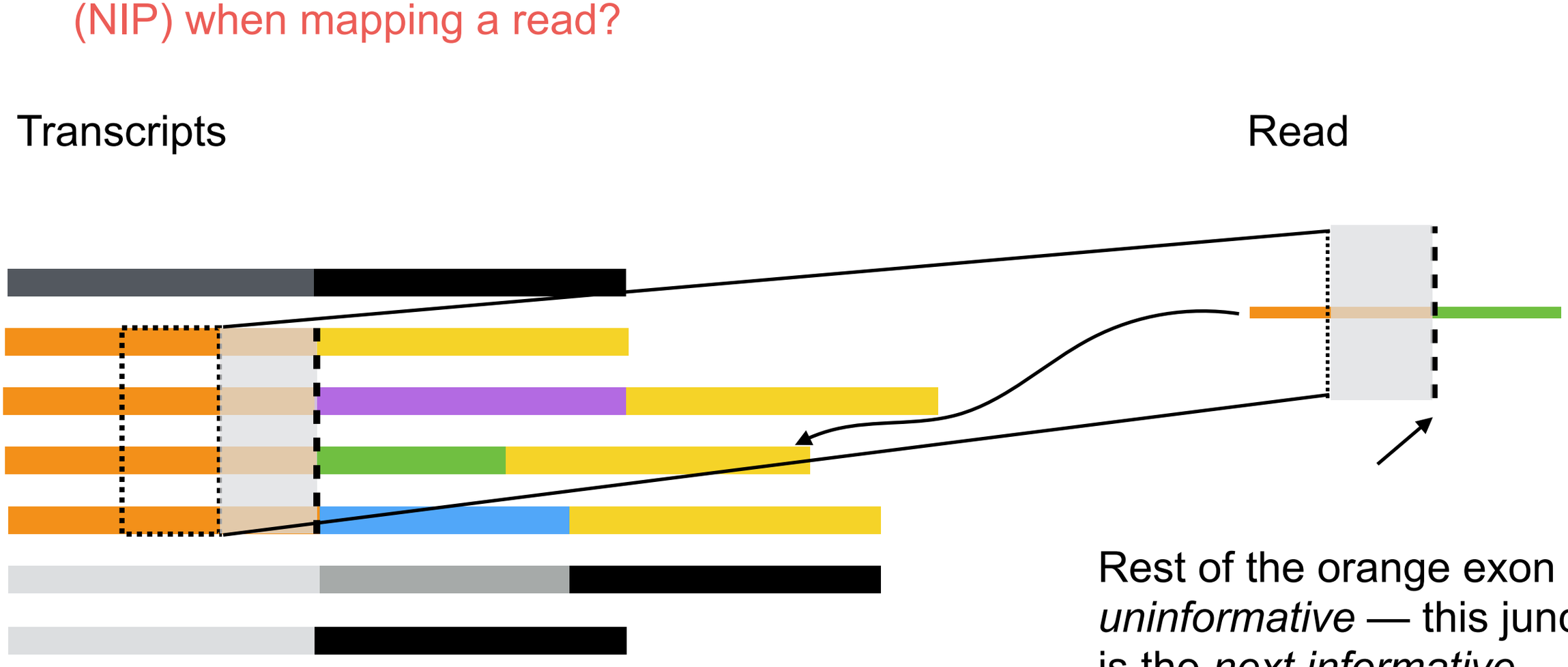


Rest of the orange exon is *uninformative* — this junction is the *next informative position*.

Mapping reads to a Transcriptome

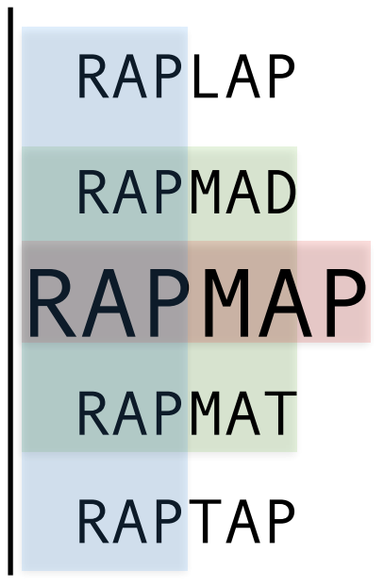
Consider the following scenario:

> Is there some ***general/formal*** way to always find the next informative position (NIP) when mapping a read?



Rest of the orange exon is *uninformative* — this junction is the *next informative position*.

RapMap: A Rapid, Sensitive and Accurate Tool for Mapping RNA-seq Reads to Transcriptomes



GitHub repository: https://github.com/COMBINE-lab/RapMap

Preprint: http://biorxiv.org/content/early/2016/01/16/029652

(appeared @ ISMB 16)

RapMap Index

Generalized suffix array on transcriptome ($ character separating transcripts)

Hash from k-mers to SA intervals (for speed) (can be **dense** or **minimum perfect hash**)

Very fast bit-vector rank — rank9*— allow constant time access to transcript start positions in generalized suffix array

Benefits of this indexing structure

The suffix array allows us to encode / find the NIPs ***dynamically*** (and guided by the length of matching context)

Allows us to efficiently deal with *intervals* of exact matches (efficient).

Length of context changes *dynamically* with quality of data (errors).

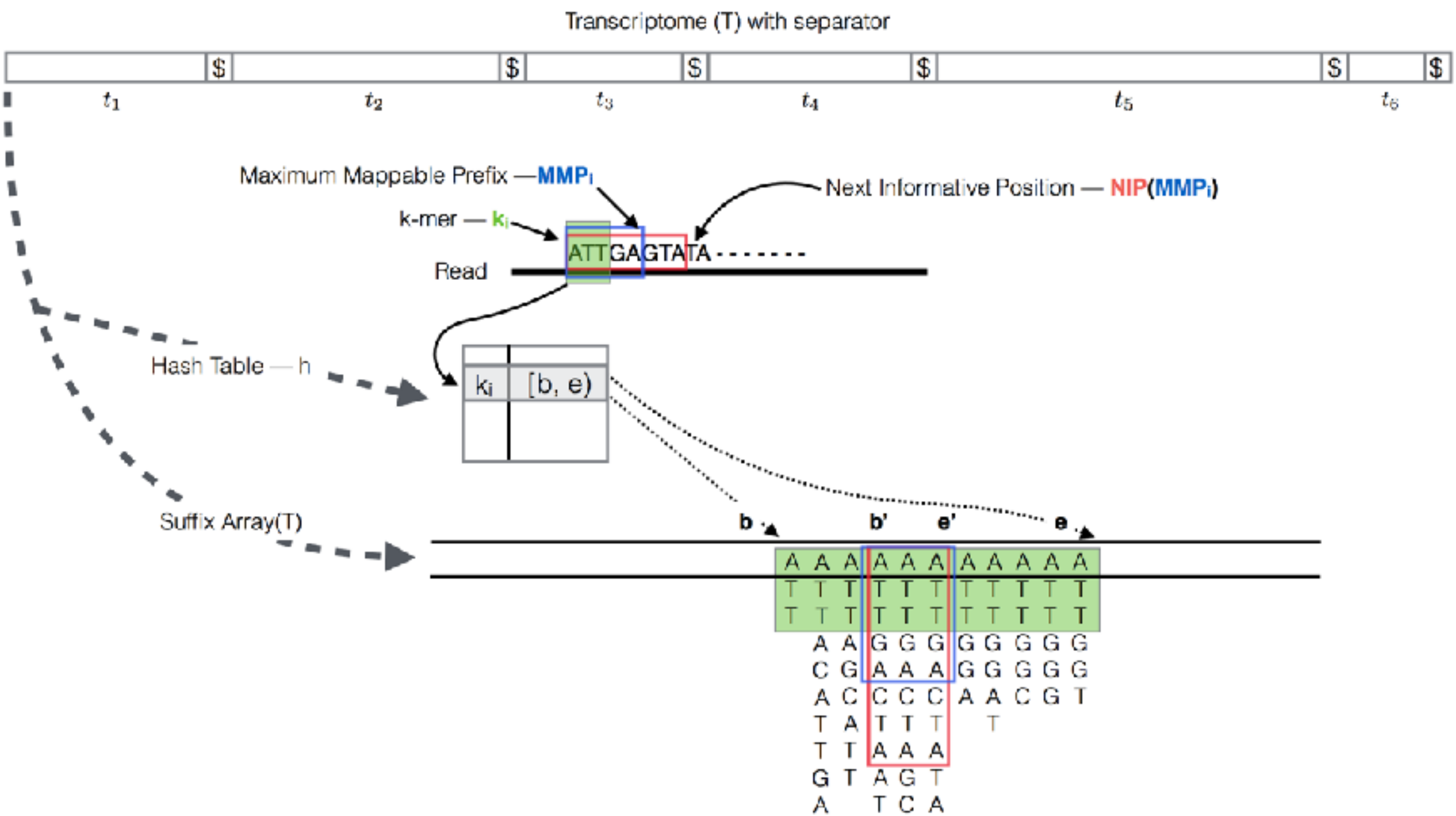
Moving from mapping to full alignment becomes very efficient (*ongoing work*).

*Sebastiano Vigna. Broadword implementation of rank/select queries. In Proc. InternationalConference on Experimental Algorithms, WEA'08, pages 154–168, 2008

An algorithm for quasi-mapping

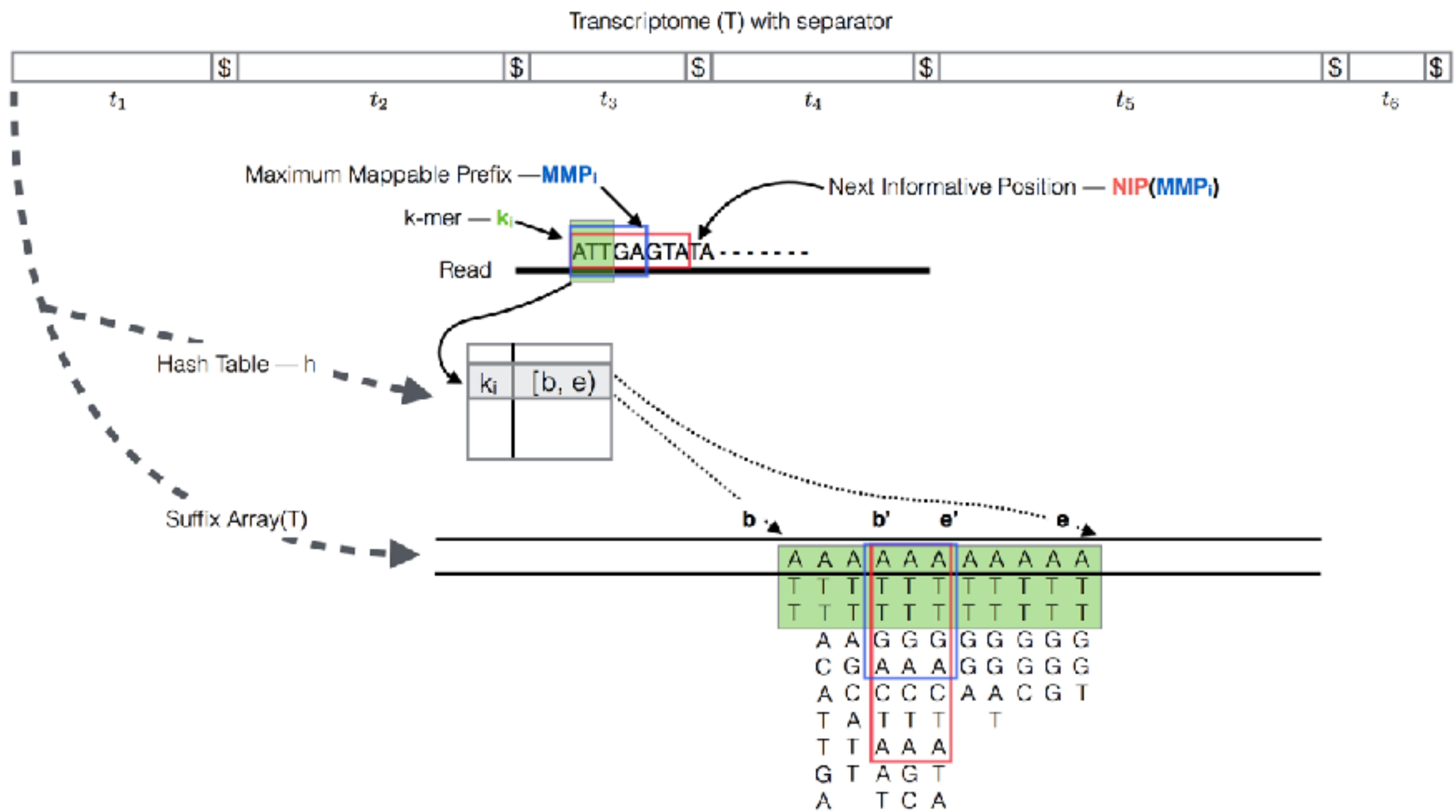
Move from left to right along read, until we find a k-mer with non-empty SA interval.

Compute Maximum Mappable Prefix (**MMP**) starting with this k-mer — logarithmic in k-mers SA interval



An algorithm for quasi-mapping

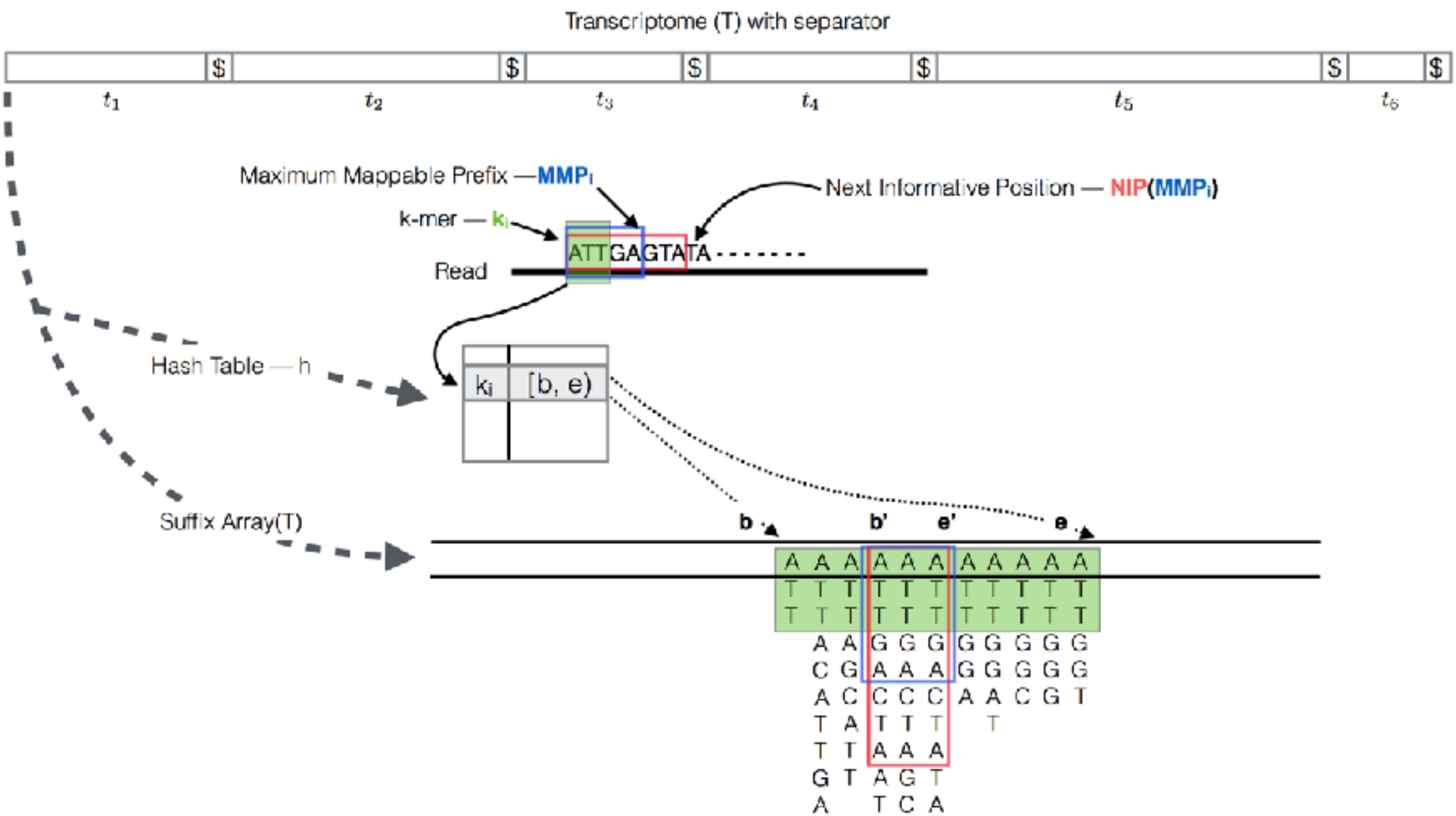
Compute **NIP** of this **MMP** — (fast) linear in read length



An algorithm for quasi-mapping

Compute **NIP** of this **MMP** — (fast) linear in read length

intuitively: **NIP** jumps you to the next exon boundary overlapping the read (need not be an actual exon boundary)



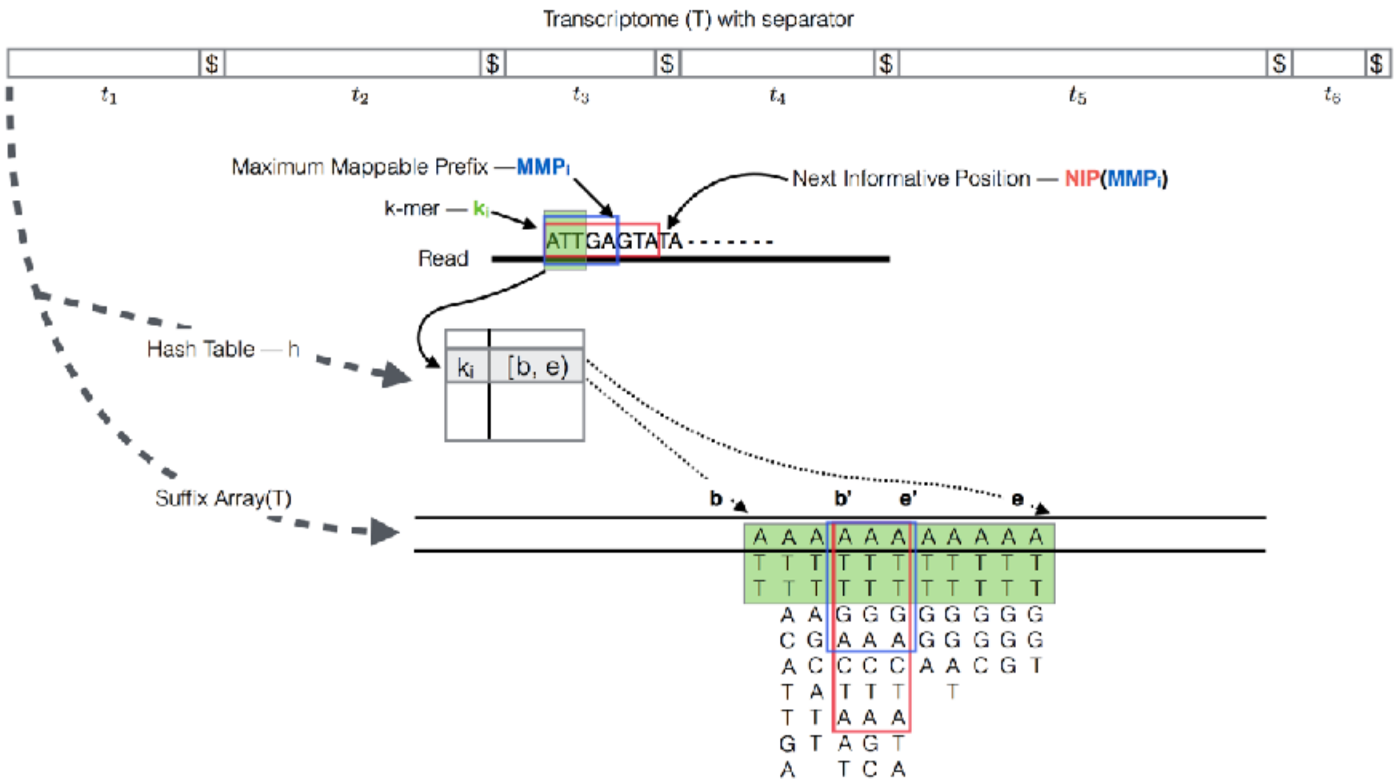
An algorithm for quasi-mapping

Produces a set of disjoint *hits* over each query (read).

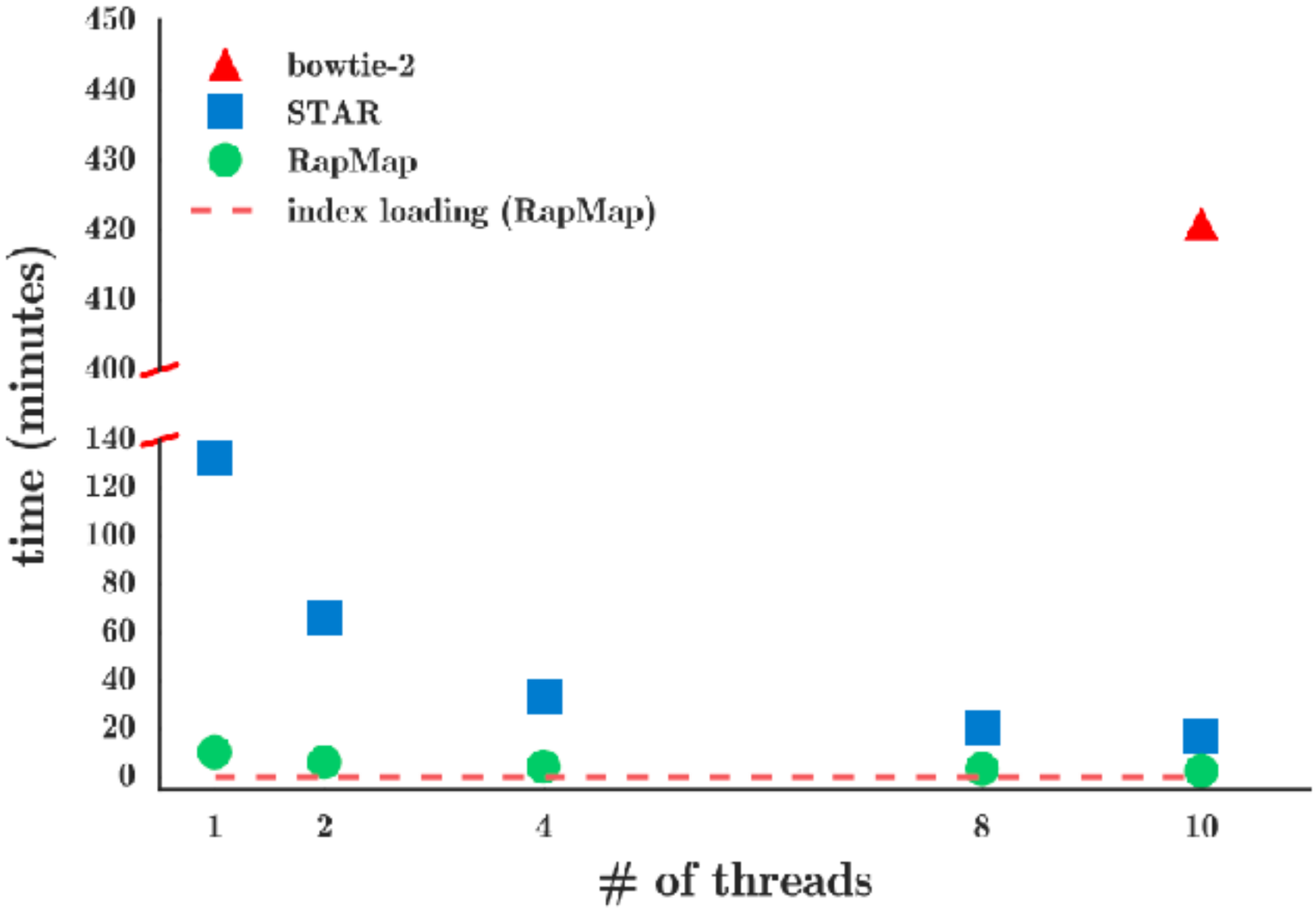
A *hit* is a tuple — (query offset, orientation, length, SA-interval)

Mappings are determined by a *consensus* mechanism over hits:

- *default:* a read maps to a transcript if that transcript appears in **every hit for that read**.
- other (stricter or looser) mechanisms are trivial to enforce (e.g. co-linearity of hits wrt read & reference).



Quasi-mapping is Fast



Can map **75 million paired-end reads** (76 bp) to the human transcriptome in matter of **minutes**; even with few threads.

Note: High degree of multi-mapping and inability to report top "**stratum**" means Bowtie2 is often reporting more than the "best" mapping (though it's commonly used in this context).

Quasi-mapping is Accurate

	Bowtie 2	Kallisto	RapMap	STAR
reads aligned	47579567	44804857	47613536	44711604
recall	97.41	91.60	97.49	91.35
precision	98.31	97.72	98.48	97.02
F1-score	97.86	94.56	97.98	94.10
FDR	1.69	2.28	1.52	2.98
hits per read	5.98	5.30	4.30	3.80

Bowtie 2: BWT-based aligner

Kallisto: dBG-based pseudoaligner

RapMap: SA-based quasi-mapper

STAR: SA-based aligner

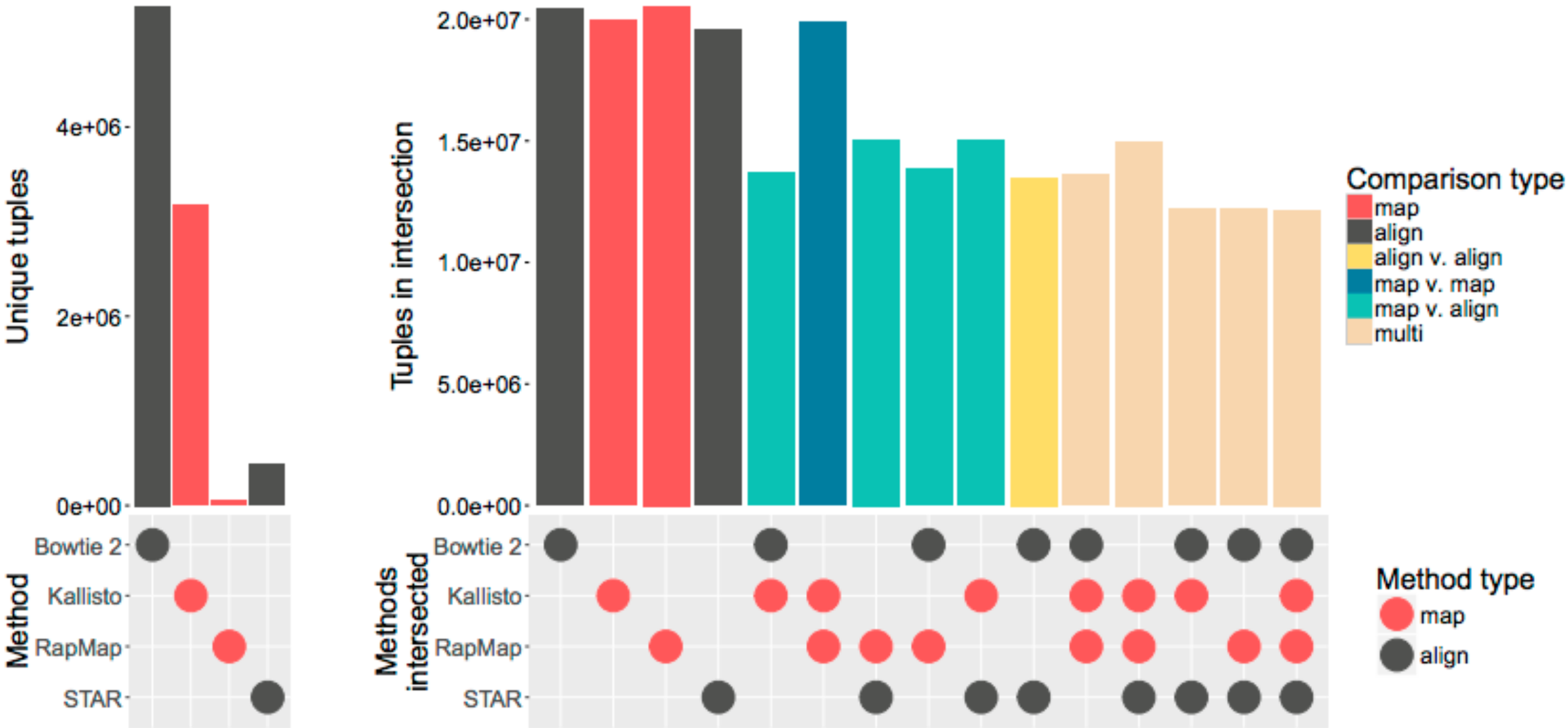
TP = True transcript of origin was in the set returned by the method

FP = Mappings were returned for the read, none of which were to the true transcript

FN = Read is un-mapped, but derives from the transcriptome

Hits per read = Avg. # of mappings returned for the reads
How many *extra* mappings did we report?

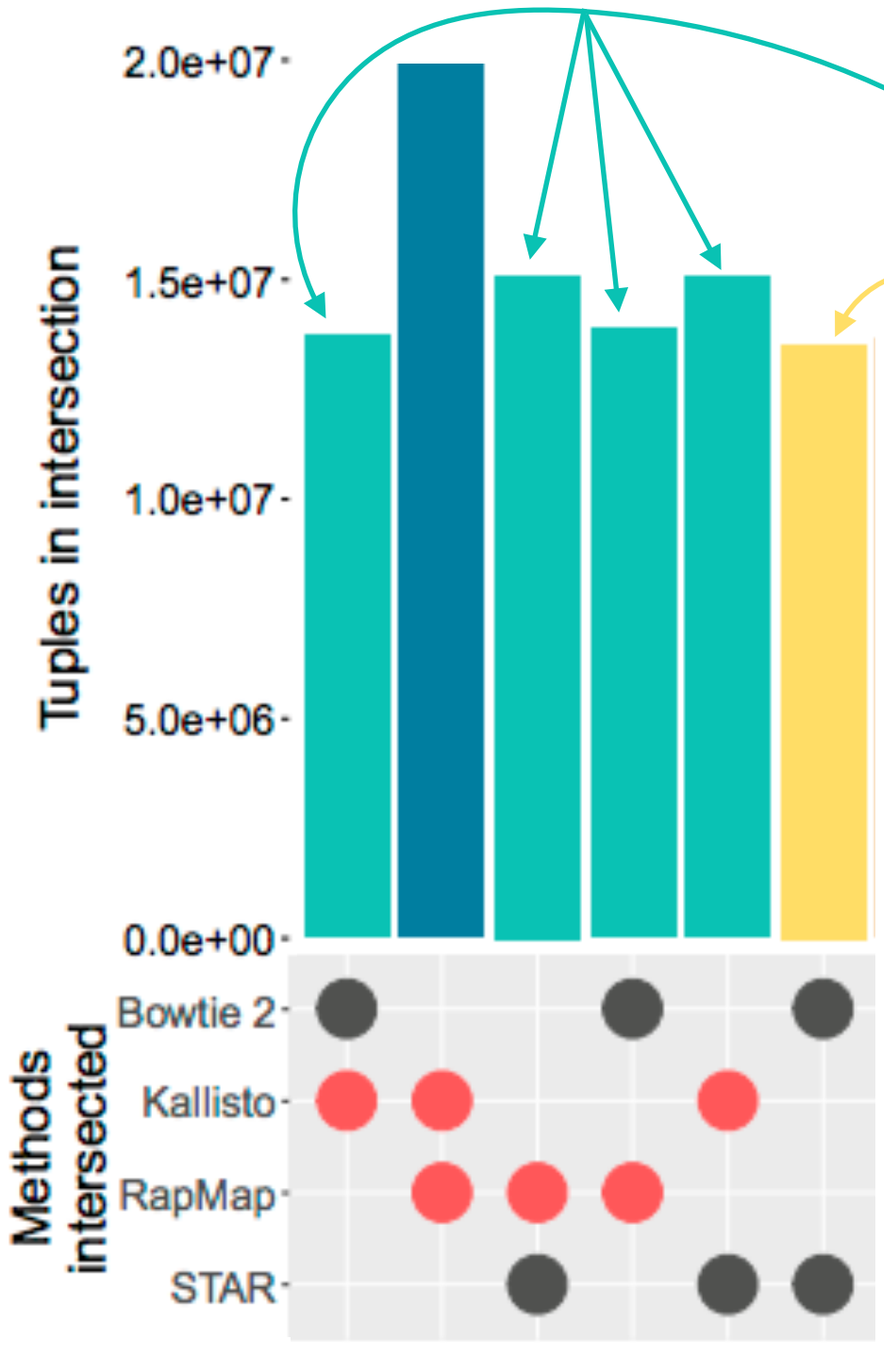
Quasi-mapping and Alignment Agree Well



A *tuple* consists of a read id and set of transcripts e.g. (r_i, {t_1, t_2, t_6})

Two methods *agree* on the mappings of a read if they return the same tuple; otherwise they disagree

Quasi-mapping and Alignment Agree Well



Mappers agree with **Aligners**, at least as often as **aligners** agree with **eachother**.

A *tuple* consists of a read id and set of transcripts e.g. (r_i, {t_1, t_2, t_6})

Two methods *agree* on the mappings of a read if they return the same tuple; otherwise they disagree

Where might we use quasi-mapping?

We believe there are ***many*** places where this replacement can be made. I'll discuss one in some depth (and mention a second):

1)Transcript-level quantification

- Determine abundance of transcripts from a collection of RNA-seq reads.
- The quasi-mapping information is sufficient to yield estimates *as accurate as full alignment*.

2)*de novo* transcript clustering

- Find groups of related contigs likely from the same transcript / gene
- Such groups help improve downstream analysis (e.g. differential expression testing)

Obviously, alignments are *necessary* for certain types of analysis (e.g. variant detection).