Toward building an automated bioinformatician: more accurate transcript assembly via parameter advising

Dan DeBlasio
dandeblasio.com
danfdeblasio

with Kwanho Kim and Carl Kingsford

slides: dandeblasio.com/AutoAlg19

Modern science is computational

Modern science is increasingly computational.

- Particularly in genomics, where experiments have multiple computational steps.
- Domain problems have in turn lead to algorithmic advances.

More domain experts are relying on computational tools.

Machine learning can help these scientists find better results.

Key problem in bioinformatics

Going to focus the transcript assembly problem

- Used to reconstruct the expressed transcripts in a sample.
- Helps in disease studies to find differences between conditions.
- One gene has multiple transcripts, each serving a different purpose.

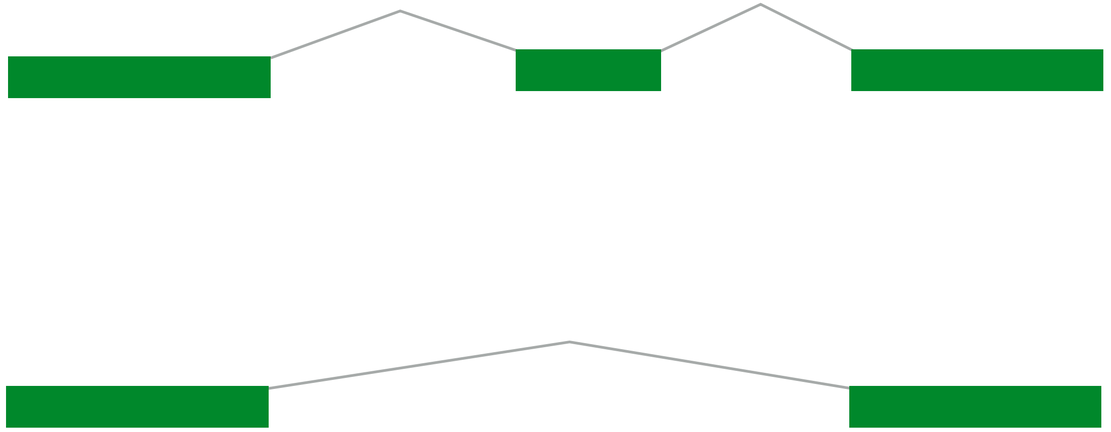
Transcript assembly (TA)

Given

- a set of RNA-seq reads aligned to a reference genome, and
- a set of thresholds for transcript construction

find:

- a set of constructed transcripts that explains the reads.



Bioinformatics software

TA and many other fundamental problems in bioinformatics are difficult.

- Many are computationally inefficient to solve exactly.
- Many tools developed for these problems.
- Each tool has many parameters whose values have an impact on the output.

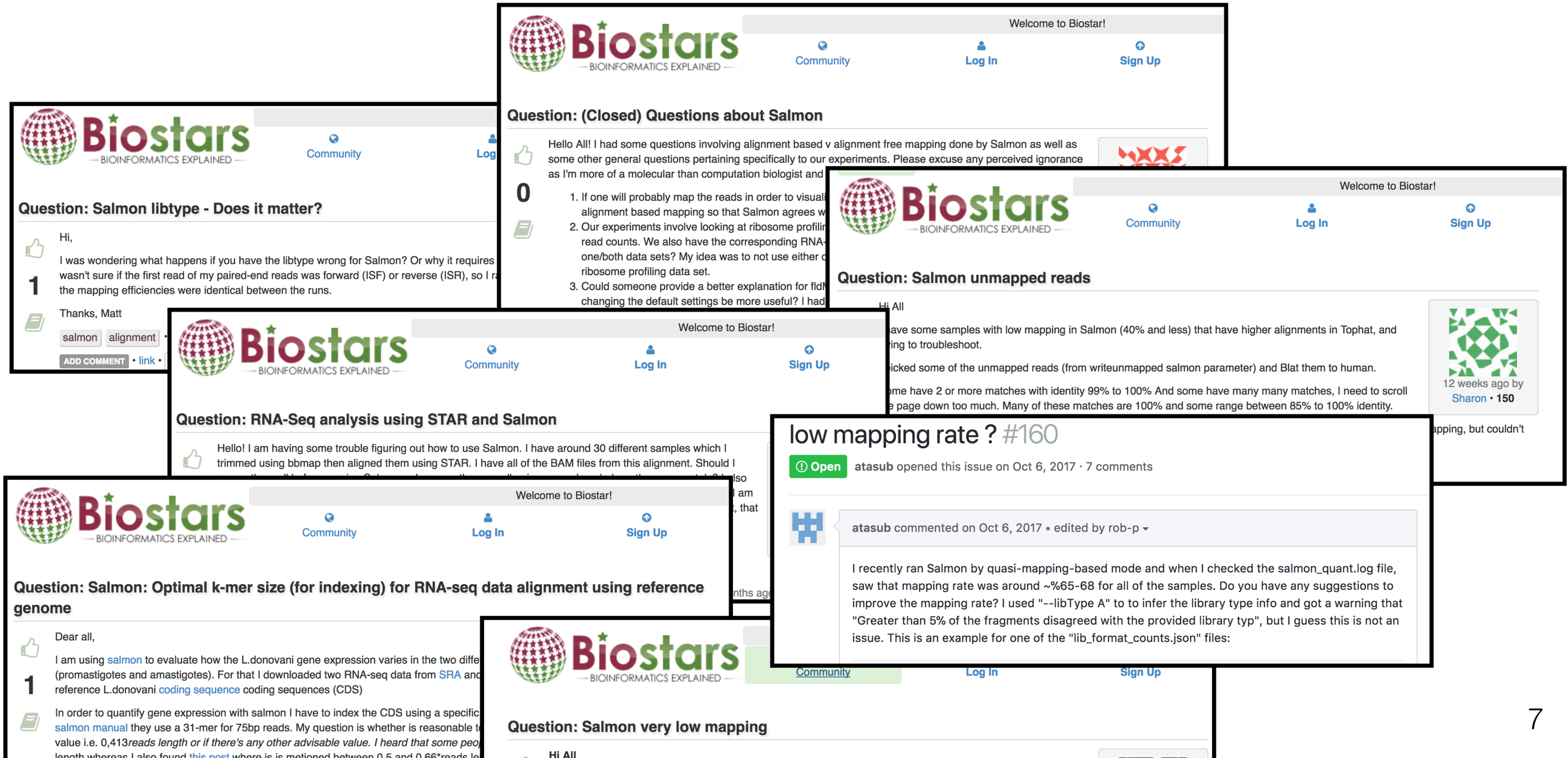
Tunable parameters

```
Quant
==========
Perform dual-phase, mapping-based estimation of
transcript abundance from RNA-seq reads

salmon quant options:


basic options:
  -v [ --version ]                      print version string
  -h [ --help ]                         produce help message
  -i [ --index ] arg                    Salmon index
  -l [ --libType ] arg                  Format string describing the library  type
  -r [ --unmatedReads ] arg             List of files containing unmated reads  of (e.g. single-end reads)
  -1 [ --mates1 ] arg                   File containing the #1 mates
  -2 [ --mates2 ] arg                   File containing the #2 mates
  -o [ --output ] arg                   Output quantification file.
  --discardOrphansQuasi                 [Quasi-mapping mode only] : Discard  orphan mappings in quasi-mapping mode.  If this flag is passed then only paired mappings
                                        will be considered toward  quantification estimates.  The default  behavior is to consider orphan mappings if no valid paired
                                        mappings exist.   This flag is independent of the option  to write the orphaned mappings to file  (--writeOrphanLinks).
  --allowOrphansFMD                     [FMD-mapping mode only] : Consider  orphaned reads as valid hits when  performing lightweight-alignment.  This option will
                                        increase sensitivity (allow more reads to map and more transcripts  to be detected), but may decrease  specificity as orphaned
                                        alignments are  more likely to be spurious.
  --seqBias                             Perform sequence-specific bias  correction.
  --gcBias                              [beta for single-end reads] Perform  fragment GC bias correction
  -p [ --threads ] arg                  The number of threads to use  concurrently.
  --incompatPrior arg                   This option sets the prior probability  that an alignment that disagrees with  the specified library type (--libType)  results
                                        from the true fragment origin.  Setting this to 0 specifies that  alignments that disagree with the  library type should be
                                        "impossible",  while setting it to 1 says that  alignments that disagree with the  library type are no less likely than  those
                                        that do
  -g [ --geneMap ] arg                  File containing a mapping of  transcripts to genes.  If this file is  provided Salmon will output both  quant.sf and
                                        quant.genes.sf files,  where the latter contains aggregated  gene-level abundance estimates.  The  transcript to gene mapping
                                        should be  provided as either a GTF file, or a in  a simple tab-delimited format where  each line contains the name of a
                                        transcript and the gene to which it  belongs separated by a tab.  The  extension of the file is used to  determine how the file
                                        should be  parsed.  Files ending in '.gtf', '.gff' or '.gff3' are assumed to be in GTF  format; files with any other extension
                                        are assumed to be in the simple format. In GTF / GFF format, the  "transcript_id" is assumed to contain  the transcript
                                        identifier and the  "gene_id" is assumed to contain the  corresponding gene identifier.
  -z [ --writeMappings ] [=arg(=-)]     If this option is provided, then the  quasi-mapping results will be written  out in SAM-compatible format.  By  default, output
                                        will be directed to  stdout, but an alternative file name  can be provided instead.
  --meta                                If you're using Salmon on a metagenomic dataset  consider setting this flag to  disable parts of the abundance  estimation model
```

Tunable parameters



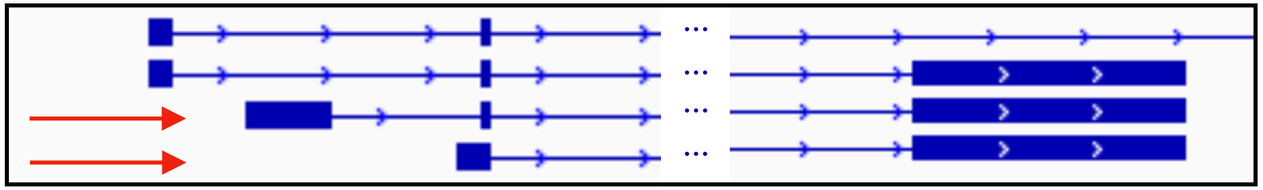
Tunable parameters

Most users rely on the default parameter settings,

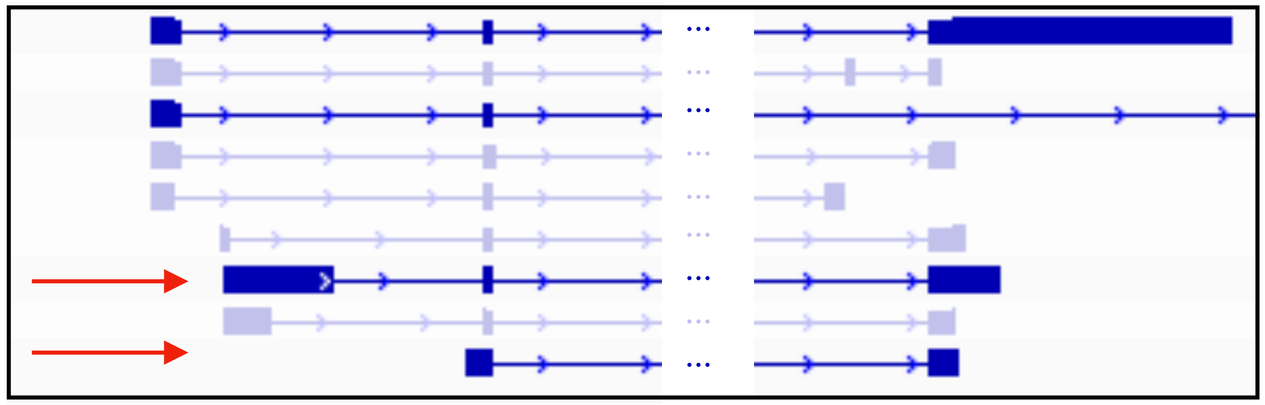
- which are meant to work well on average,
- but the most interesting examples are not typically "average".

Default Parameter Vector

Optimized Parameter Vector



Reference Transcriptome



The default parameter choices miss two transcripts that are supported by the data and in the reference transcriptome.

Tunable parameters

It's not just a problem in computational biology!

Journal of Artificial Intelligence Research 36 (2009) 267-306 Submitted 06/09; published 10/09

SATzilla: Portfolio-based Algorithm Selection for SAT

Lin Xu XULIN730@CS.UBC.CA
Frank Hutter HUTTER@CS.UBC.CA
Holger H. Hoos HOOS@CS.UBC.CA
Kevin Leyton-Brown KEVINLB@CS.UBC.CA
Department of Computer Science
University of British Columbia
201-2366 Main Mall, BC V6T 1Z4, CANADA

ParamILS: An Automatic Algorithm Configuration Framework

Frank Hutter HUTTER@CS.UBC.CA
Holger H. Hoos HOOS@CS.UBC.CA
Kevin Leyton-Brown KEVINLB@CS.UBC.CA
University of British Columbia, 2366 Main Mall
Vancouver, BC, V6T1Z4, Canada

Thomas Stützle STUETZLE@ULB.AC.BE
Université Libre de Bruxelles, CoDE, IRIDIA
Av. F. Roosevelt 50 B-1050 Brussels, Belgium

Concertio

Home Solutions Blog

Concertio Launches Optimizer Studio to Help Performance Engineers and IT Professionals Achieve Peak System Performance

by admin | Feb 22, 2018 | News | 0 comments

Swarm and Evolutionary Computation 1 (2011) 19–31

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Invited paper

Parameter tuning for configuring and analyzing evolutionary algorithms

A.E. Eiben *, S.K. Smit [1]
Department of Computer Science, Vrije Universiteit Amsterdam De Boelelaan 1081a 1081 HV, Amsterdam, Netherlands

ARTICLE INFO ABSTRACT

Automated bioinformatician

Almost all pieces of scientific software have tunable parameters.

- Their settings can greatly impact the quality of output.
- Default parameters are best on average but may be bad in general.
- Mis-configuration can lead to missed or incorrect conclusions.

Can we remove parameter choice
as a source of error
in transcriptome analysis?

Advising paradigms

A priori advising looks at the input to make parameter decisions.

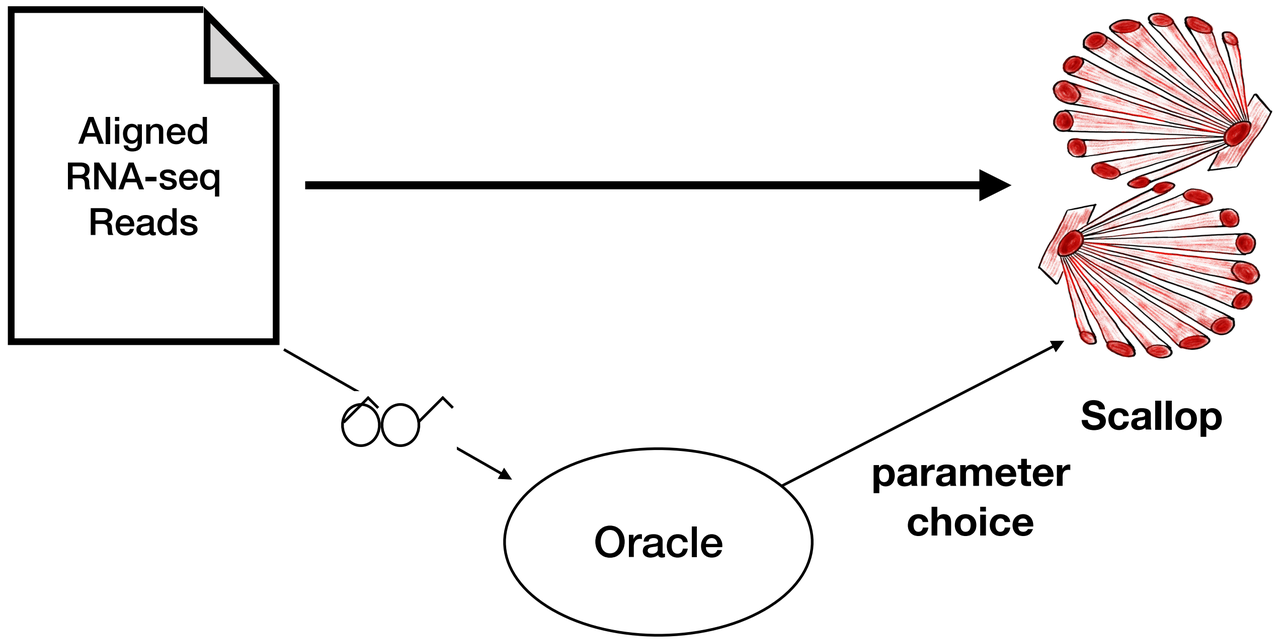
- Needs to know about the algorithm.
- Analyzes features of the particular instance.

A posteriori advising looks at program outputs to make parameter decisions.

- Has access to more information.
- Does not need to know anything about the parameters functions.

Automated bioinformatician

The goal is to find the parameter choice for a given input.



A posteriori advising

In machine learning, this is the hyper-parameter tuning problem.

- coordinate ascent
- simulated annealing
- bayesian inference
- etc.

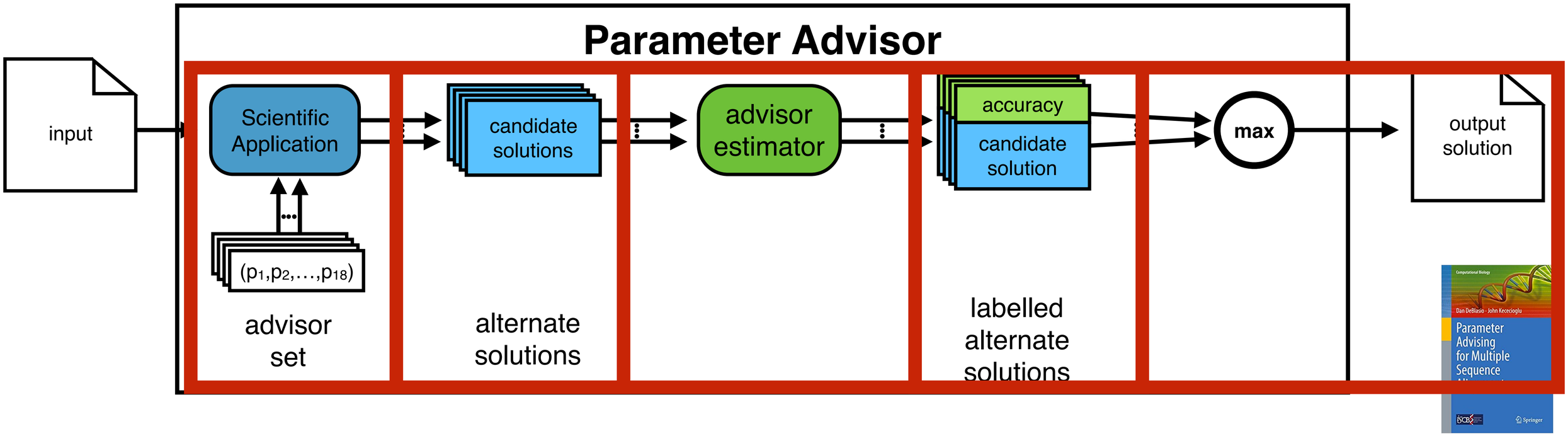
Issue is that running time is increased greatly.

- The application needs to be run multiple times.
- Those instances need to be (somewhat) sequential.

Parameter advising framework

Steps of advising:

- An advisor set of parameter choice vectors is used to obtain candidates.
- Solutions are ranked based on the accuracy estimation.
- The highest ranked candidate is returned.



Parameter advising framework

Steps of advising:

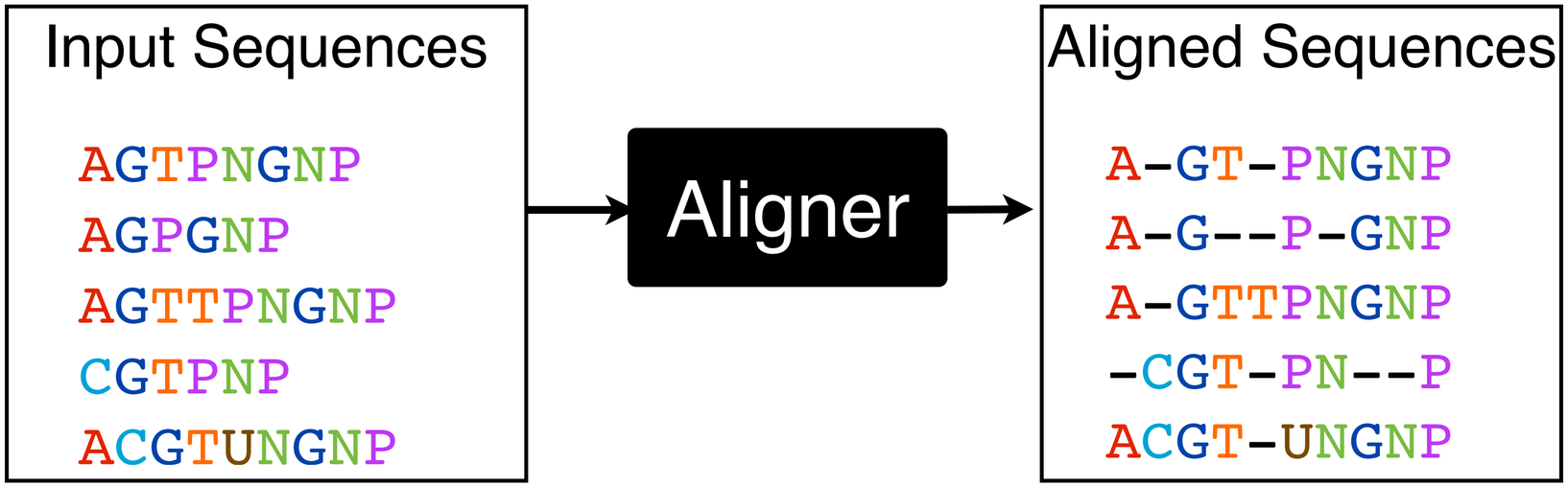
- An advisor set of parameter choice vectors is used to obtain candidates.
- Solutions are ranked based on the accuracy estimation.
- The highest ranked candidate is returned.

[DeBlasio and Kececioglu, Springer International, 2017]

Multiple sequence alignment

A fundamental problem in bioinformatics.

- NP-Complete
- many popular aligners
- many parameters whose values affect the output
- no standard metric for measuring accuracy without ground truth

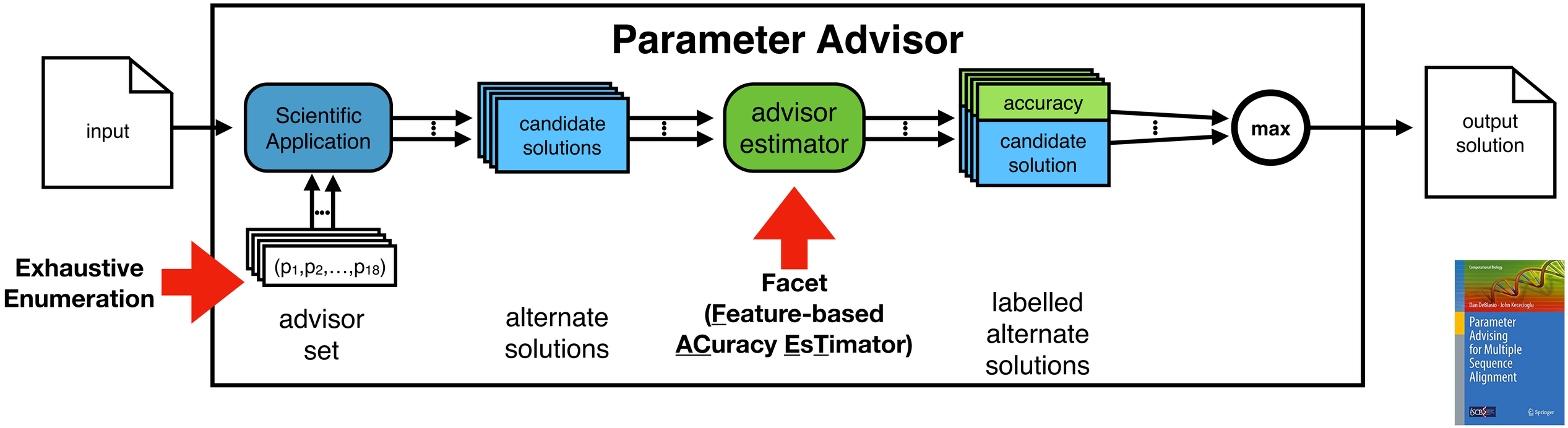


[DeBlasio and Kececioglu, Springer International, 2017]

Parameter advising framework

Steps of advising:

- An advisor set of parameter choice vectors is used to obtain candidates.
- Solutions are ranked based on the accuracy estimation.
- The highest ranked candidate is returned.

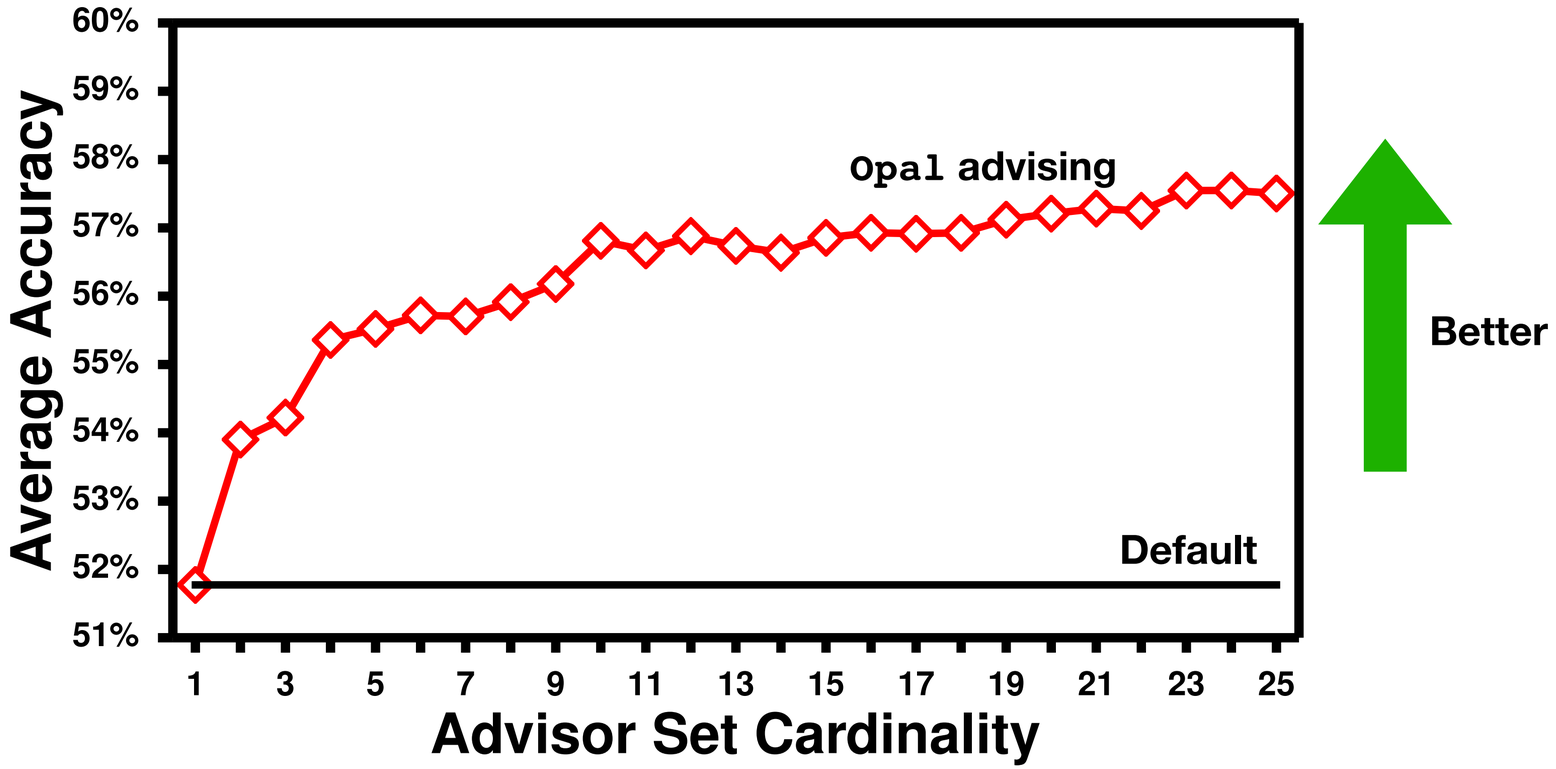


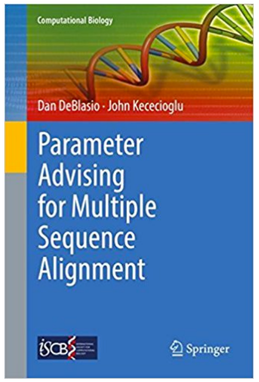
[DeBlasio and Kececioglu, Springer International, 2017]

Parameter advising

Increases accuracy for multiple sequence alignment by

- choosing a parameter choice for each input and
- accuracy increases with advisor set size, but
- so does the resource requirement.



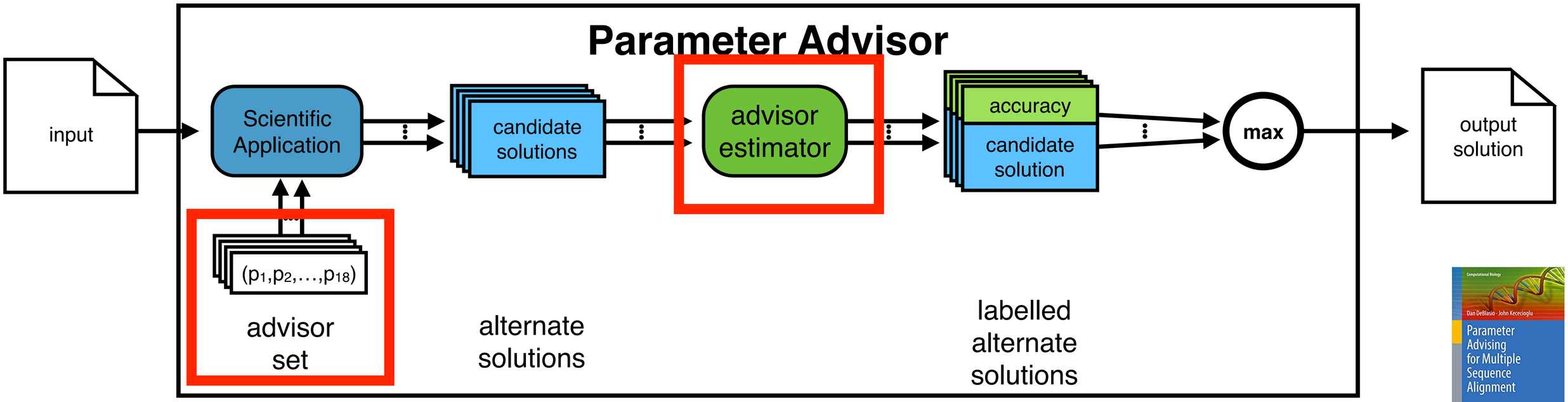


[DeBlasio and Kececioglu, Springer International, 2017]

Parameter advising framework

Components of an advisor:

- An advisor set of parameter choice vectors.
- An advisor estimator to rank solutions.



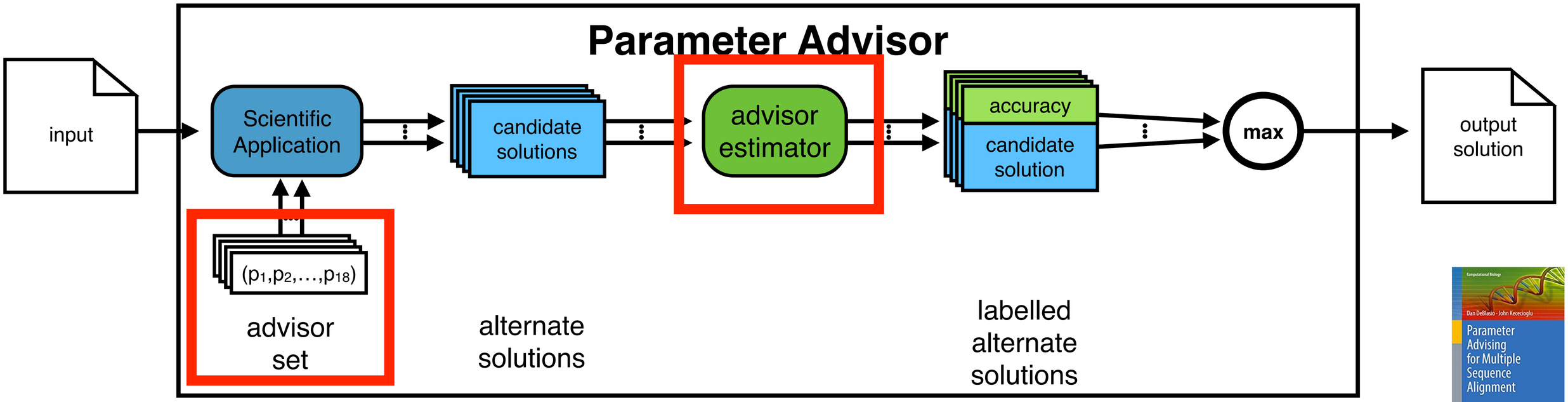
[DeBlasio and Kececioglu, Springer International, 2017]

Parameter advising framework

Components of an advisor:

- An advisor set of parameter choice vectors.
- An advisor estimator to rank solutions.

A good advisor set:
- Small
- Representative



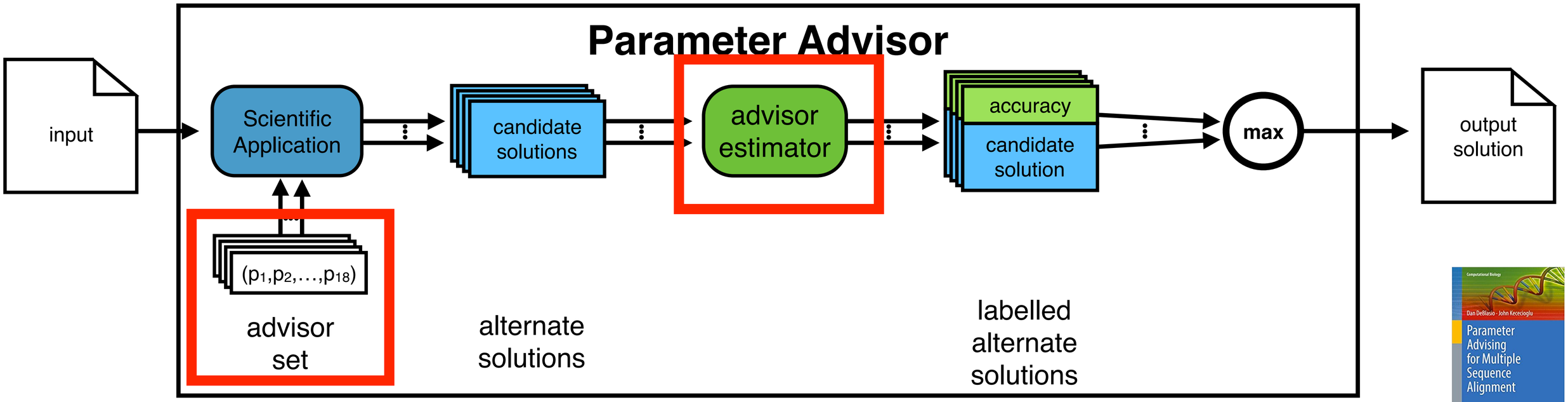
[DeBlasio and Kececioglu, Springer International, 2017]

Parameter advising framework

Components of an advisor:

- An advisor set of parameter choice vectors.
- An advisor estimator to rank solutions.

A good advisor estimator:
- Efficient
- Rank Solutions Well



[DeBlasio and Kececioglu, Springer International, 2017]

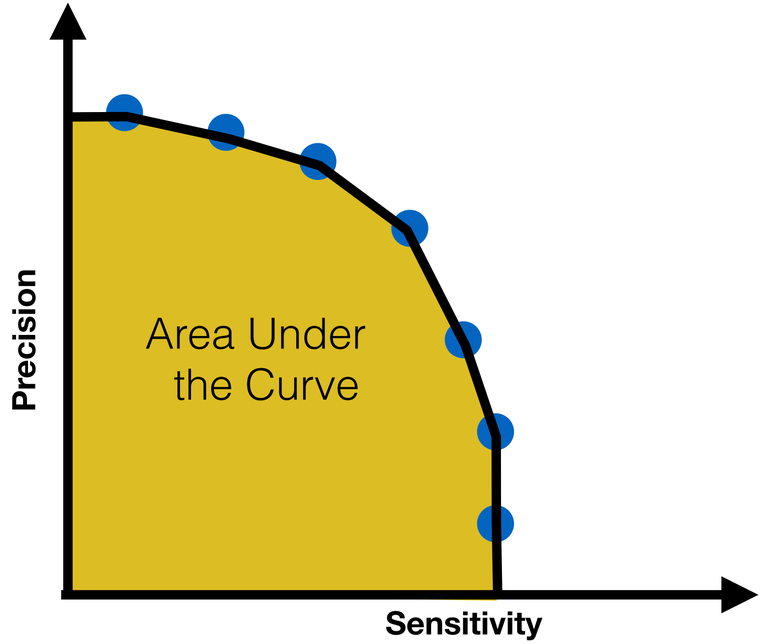
Transcript assembly

For the human genome there is a reference transcriptome.

- Contains a large set of biologically verified transcripts.
- More than will be seen in a single experiment.
- Missing novel transcripts for any given experiment.

Area Under the Curve (AUC) can be calculated using the reference transcriptome.

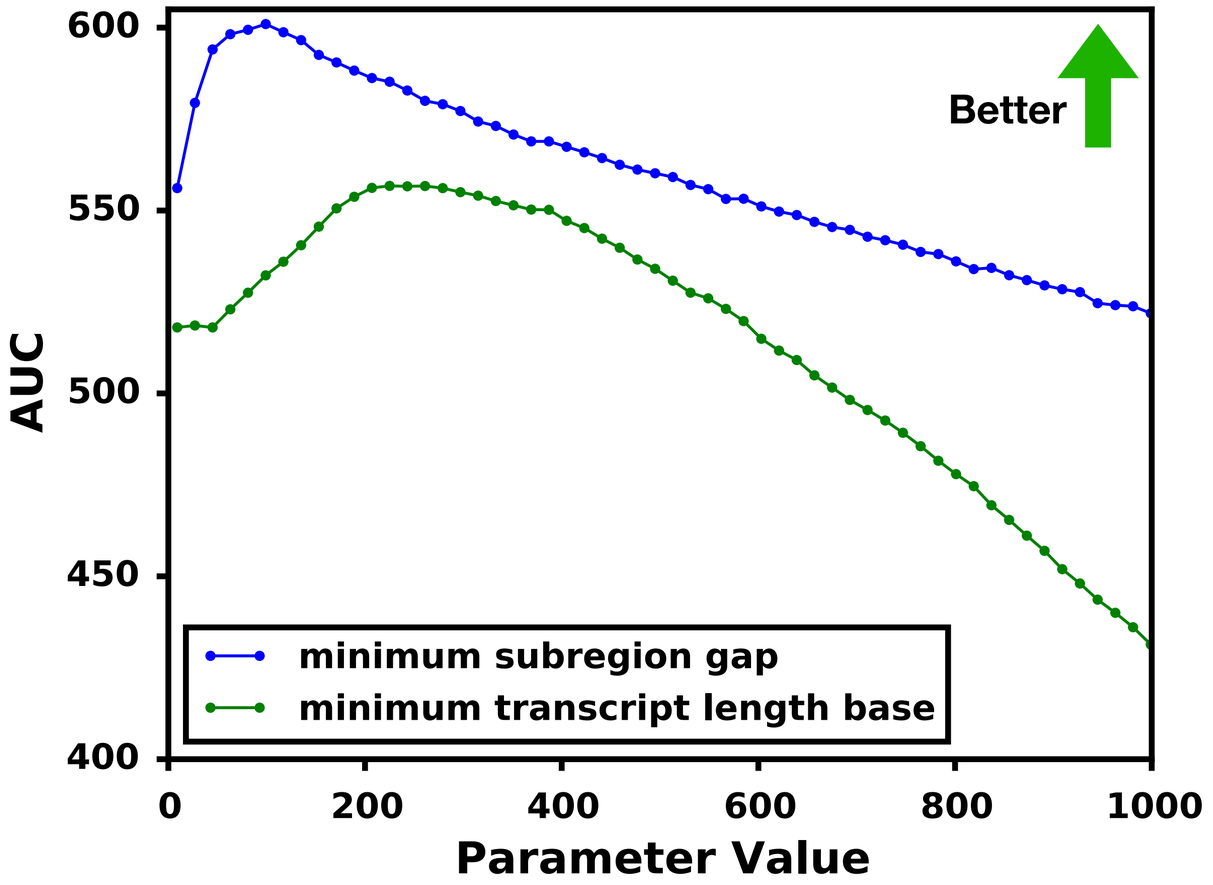
- Map assembled transcripts to the reference.
- Threshold the quality score from the assembler to get precision/sensitivity.
- Commonly used to compare assembler quality.



Scallop advising

Cannot test all combinations of parameter values.

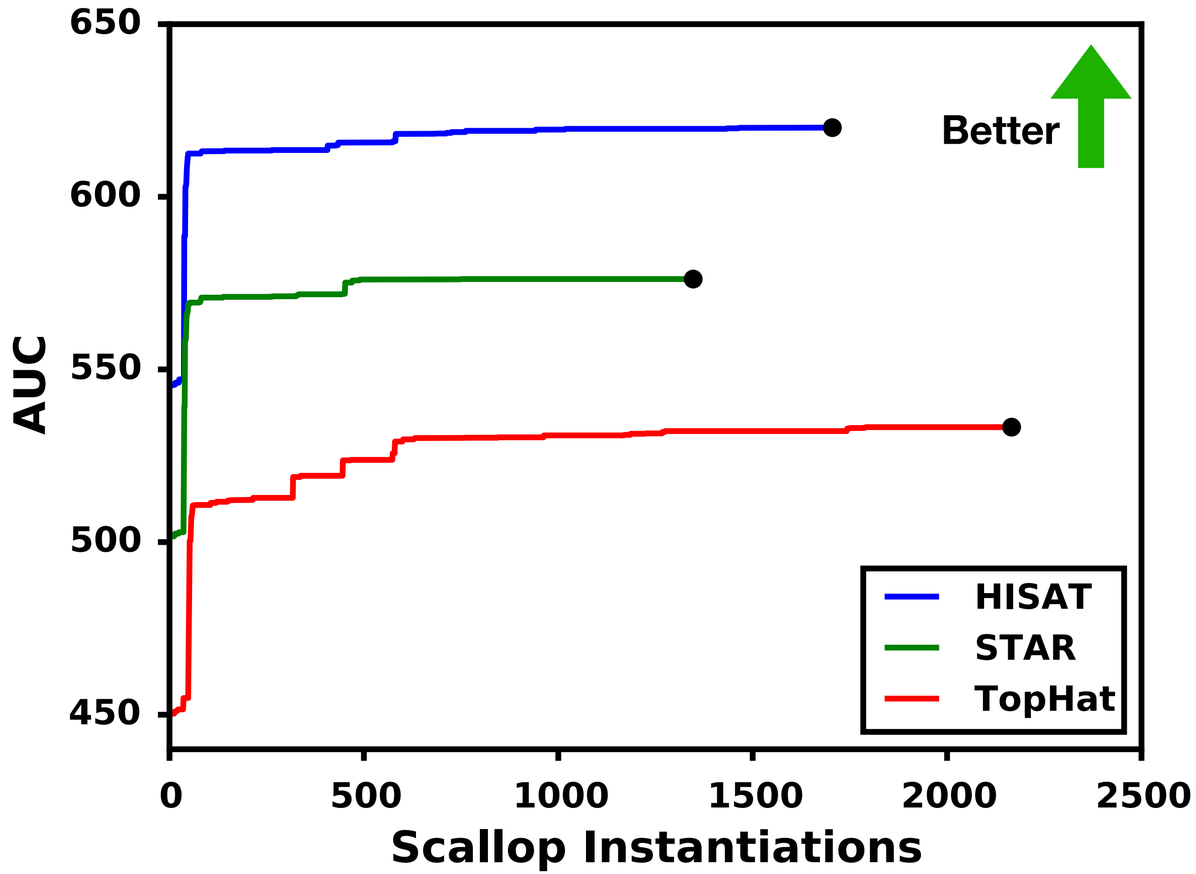
- Tested the behavior of each parameter in isolation.
- Each parameter had a single global maximum on the large regions tested.
- In general, we did not see non-global local maxima.



Scallop advising

Parameter curve smoothness means

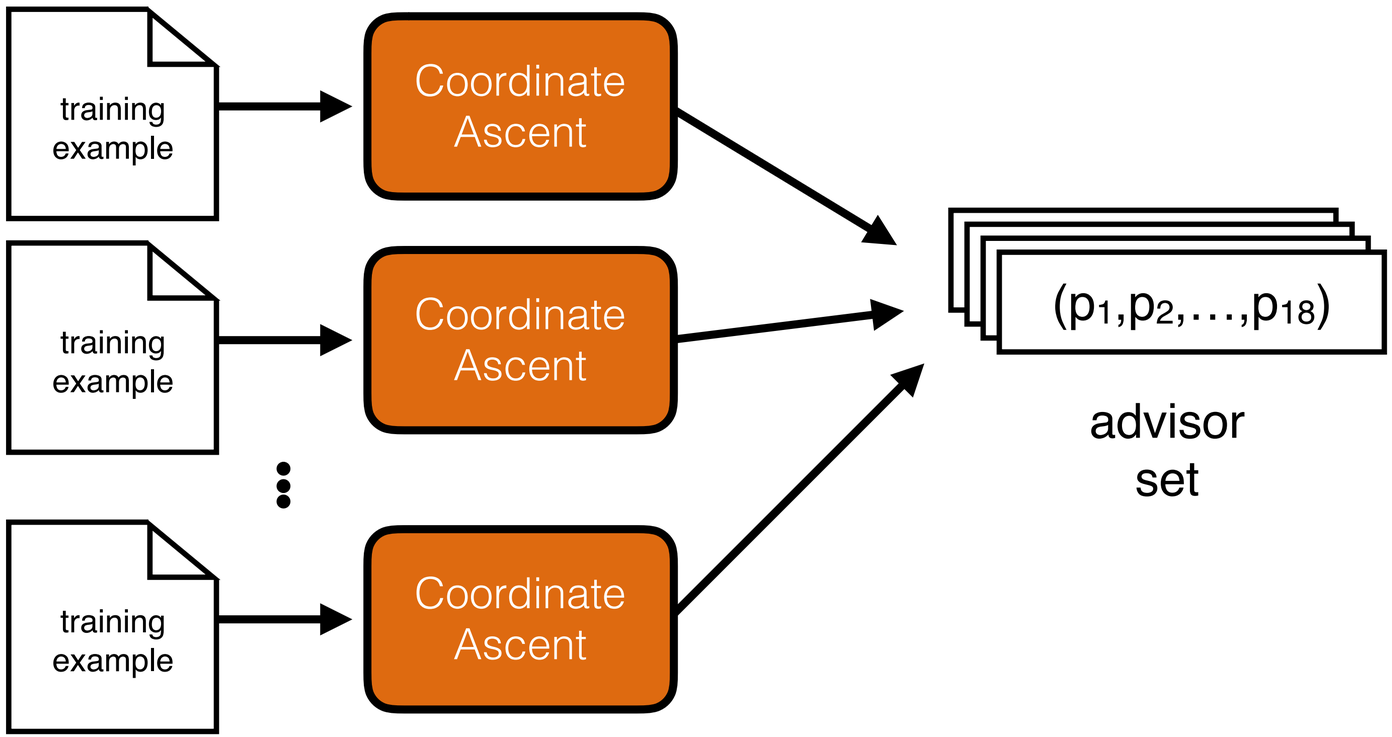
- coordinate ascent will work well
- but is slow since Scallop's running time is significant.



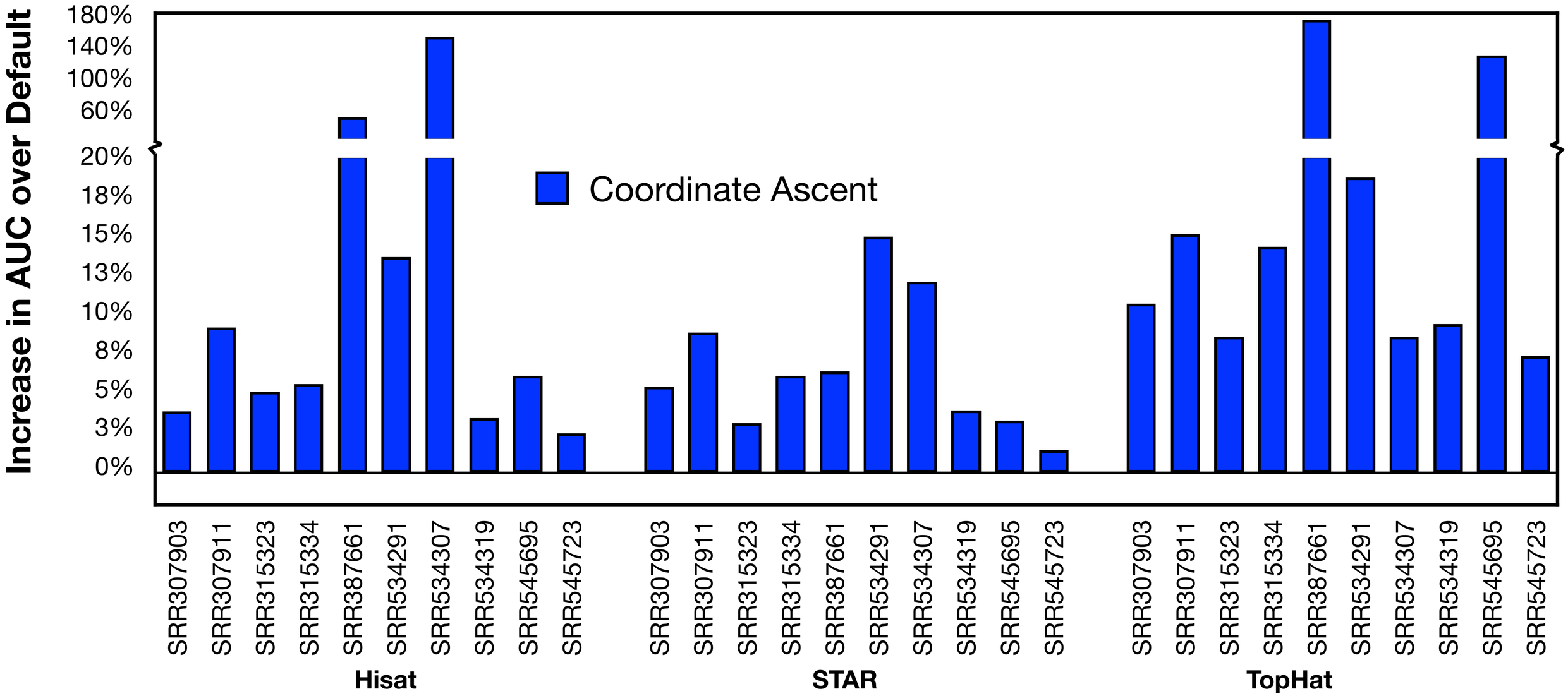
Finding an advisor set

We can use coordinate ascent to find optimal parameter vectors.

- Training samples should cover the range of expected input.
- Settings are found for all 18 tunable parameters.
- Collection of produced vectors is advisor set.
- The set is precomputed and doesn't impact the advising time.



Scallop advising

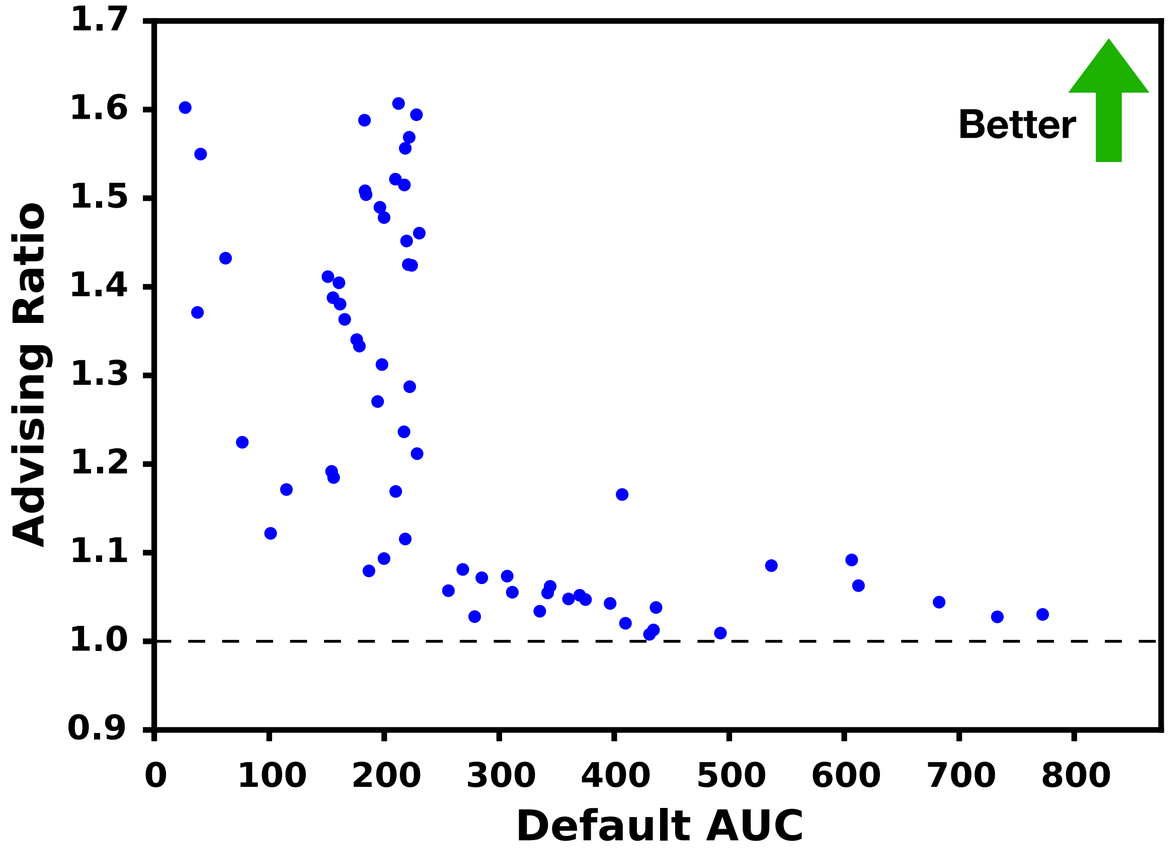


Average of 18.1% increase in AUC using Coordinate Ascent

Scallop advising

65 ENCODE dataset

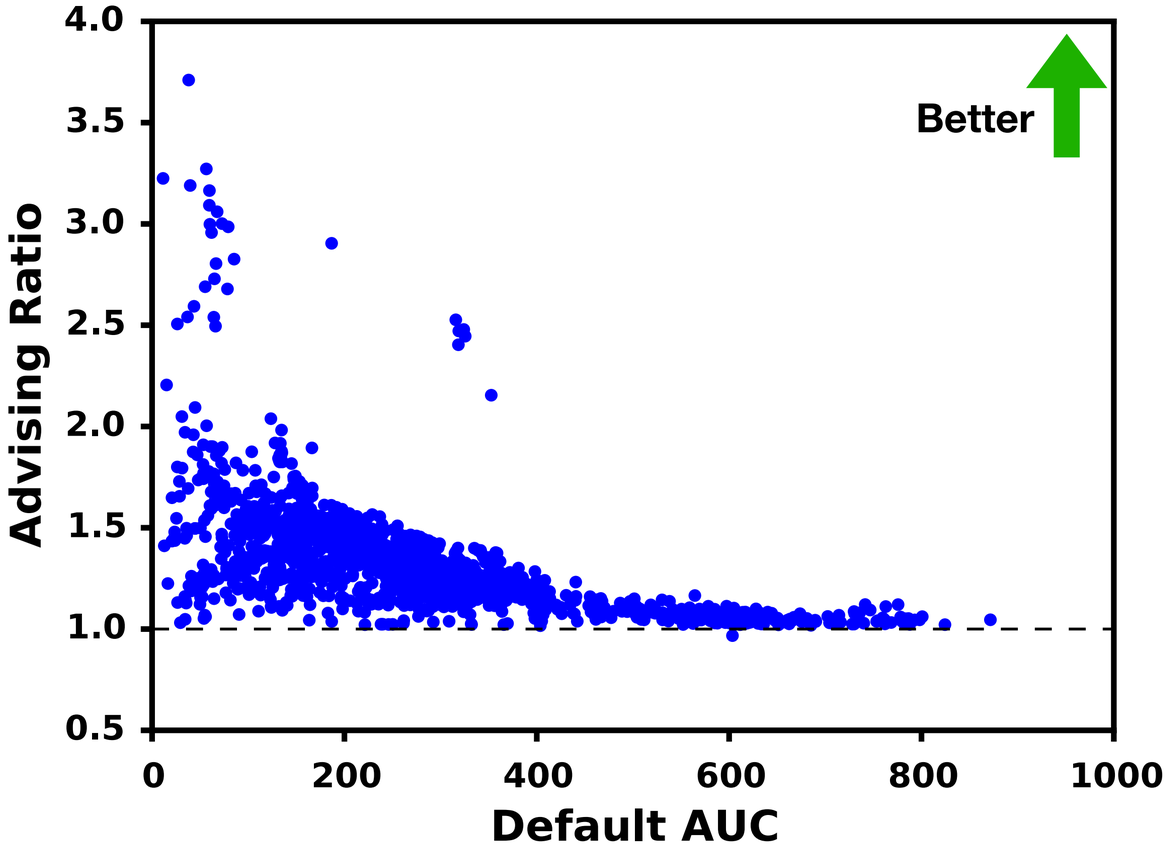
- all of the aligned RNA-seq experiments from ENCODE
- aligned using a variety of aligners
- using either the current or legacy reference genome
- stands in for the performance of advising on generic input
- **average 25.7% increase in AUC**



Scallop advising

SRA dataset

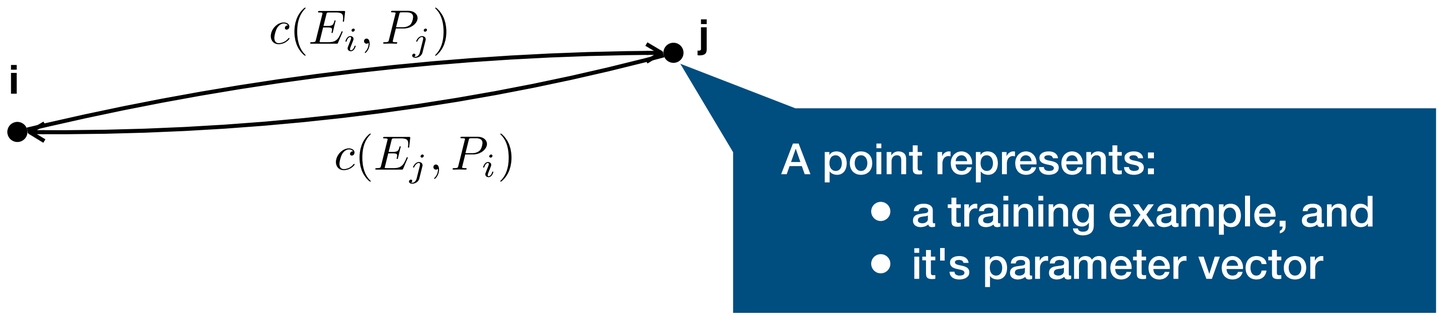
- all 1595 RNA-Seq experiments from the SRA
- aligned using STAR to the same reference genome
- represents performance of advising in a high-throughput experiment
- **average of 38.2% increase in AUC**



Advisor sub-sets

31 parameters may be too many to run in parallel

- parameter subsets were found using the *oracle set* method for advising
- parameters are meant to cover the range of inputs



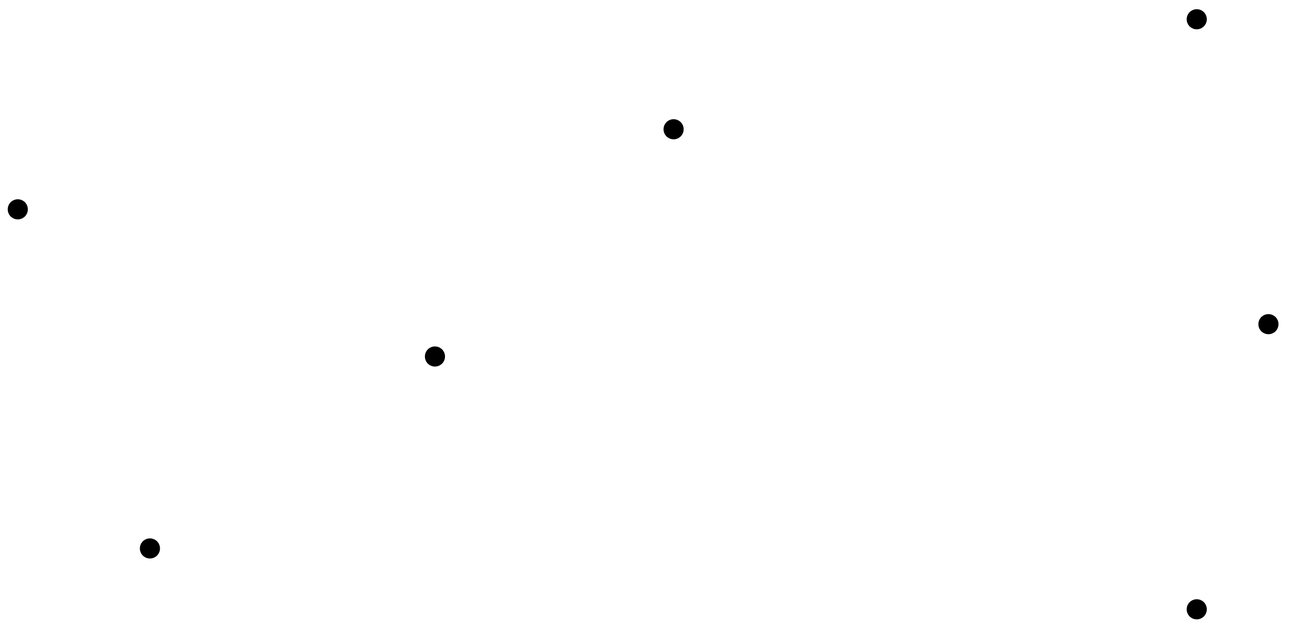
[DeBlasio and Kececioglu, Springer International, 2017]

$$c(e,p) := AUC(Scallop_{p(e)}(e)) - AUC(Scallop_p(e))$$

Advisor sub-sets

31 parameters may be too many to run in parallel

- parameter subsets were found using the *oracle set* method for advising
- parameters are meant to cover the range of inputs

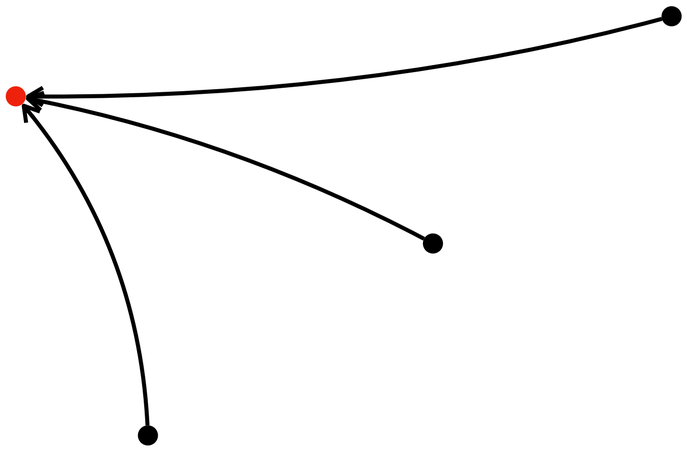


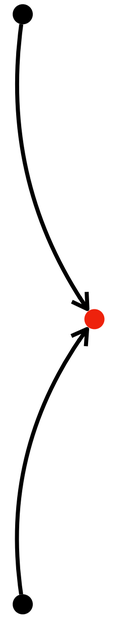
$$c(e,p) := AUC(Scallop_{p(e)}(e)) - AUC(Scallop_p(e))$$

Advisor sub-sets

31 parameters may be too many to run in parallel

- parameter subsets were found using the *oracle set* method for advising
- parameters are meant to cover the range of inputs





Find $S \subseteq \{1...n\}, |S| = k$

To minimize $\sum_i \min_{j \in S} c(E_i, P_j)$

$$c(e,p) := AUC(Scallop_{p(e)}(e)) - AUC(Scallop_p(e))$$

Advisor sub-sets

31 parameters may be too many to run in parallel

- parameter subsets were found using the *oracle set* method for advising
- parameters are meant to cover the range of inputs

Table Parameter vector subsets

Experiment/Aligner	Subset size 1	2	4	8
SRR545723/TopHat	X		X	X
SRR534291/TopHat				X
SRR387661/TopHat		X		
SRR534307/TopHat		X		
SRR387661/HISAT				X
SRR545695/HISAT				X
SRR534307/HISAT			X	X
SRR307911/STAR			X	
SRR315334/STAR				X
SRR534319/STAR				X
SRR534307/STAR			X	X

Advisor sub-sets

31 parameters may be too many to run in parallel

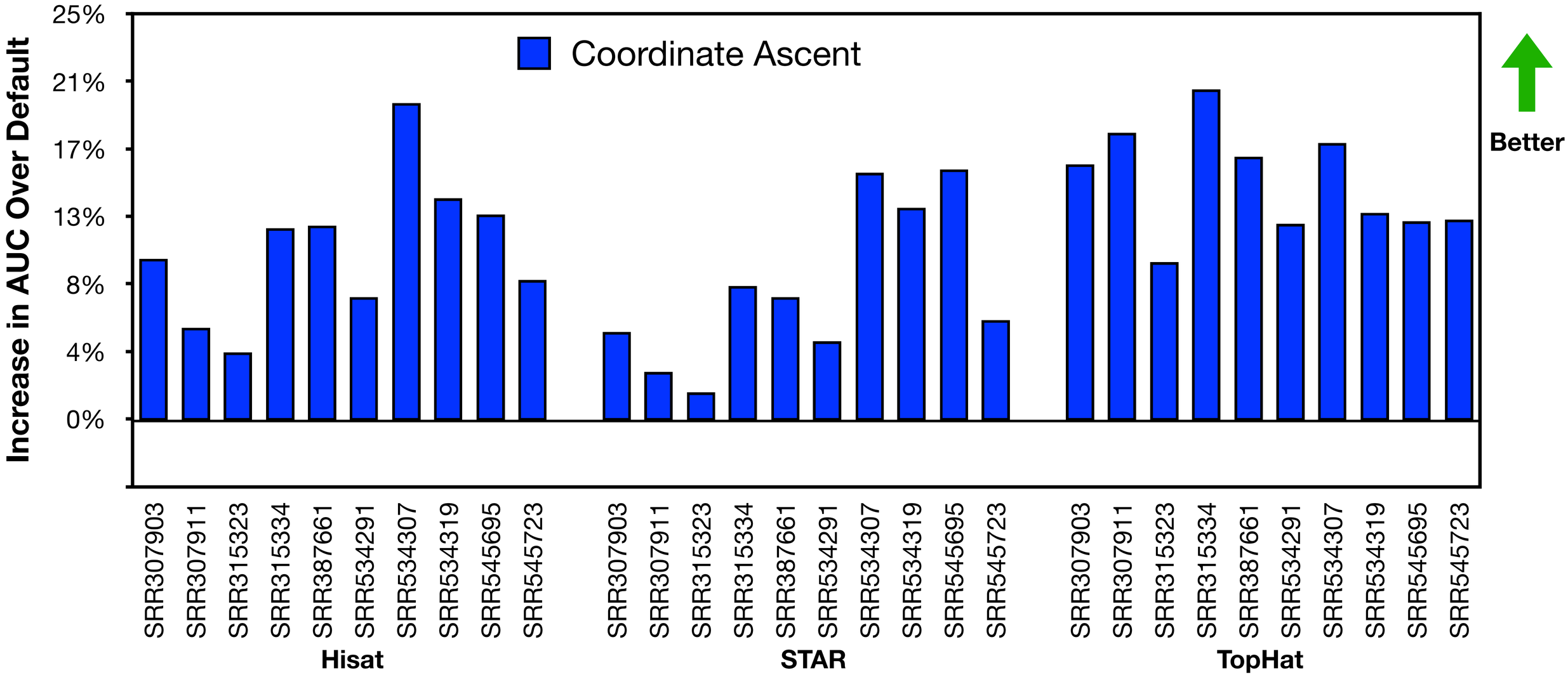
- parameter subsets were found using the *oracle set* method for advising
- parameters are meant to cover the range of inputs

Not all parameters are used when available

Table ENCODE65 Parameter Use

Experiment/Aligner	Subset size 31	8	4	2
SRR545723/TopHat	0	10.8%	38.5%	
SRR534291/TopHat	3.1%	33.8%		
SRR387661/TopHat	0			92.3%
SRR534307/TopHat	0			7.7%
SRR315323/TopHat	21.5%			
SRR307903/TopHat	7.7%			
SRR315334/TopHat	4.6%			
SRR534319/TopHat	1.5%			
SRR387661/HISAT	7.7%	10.8%		
SRR545695/HISAT	0	0		
SRR534307/HISAT	0	0	0	
SRR534319/HISAT	1.5%			
SRR307911/STAR	0		61.5%	
SRR315334/STAR	4.6%	44.6%		
SRR534307/STAR	0	0	0	
SRR534319/STAR	0	0		
SRR534291/STAR	47.7%			

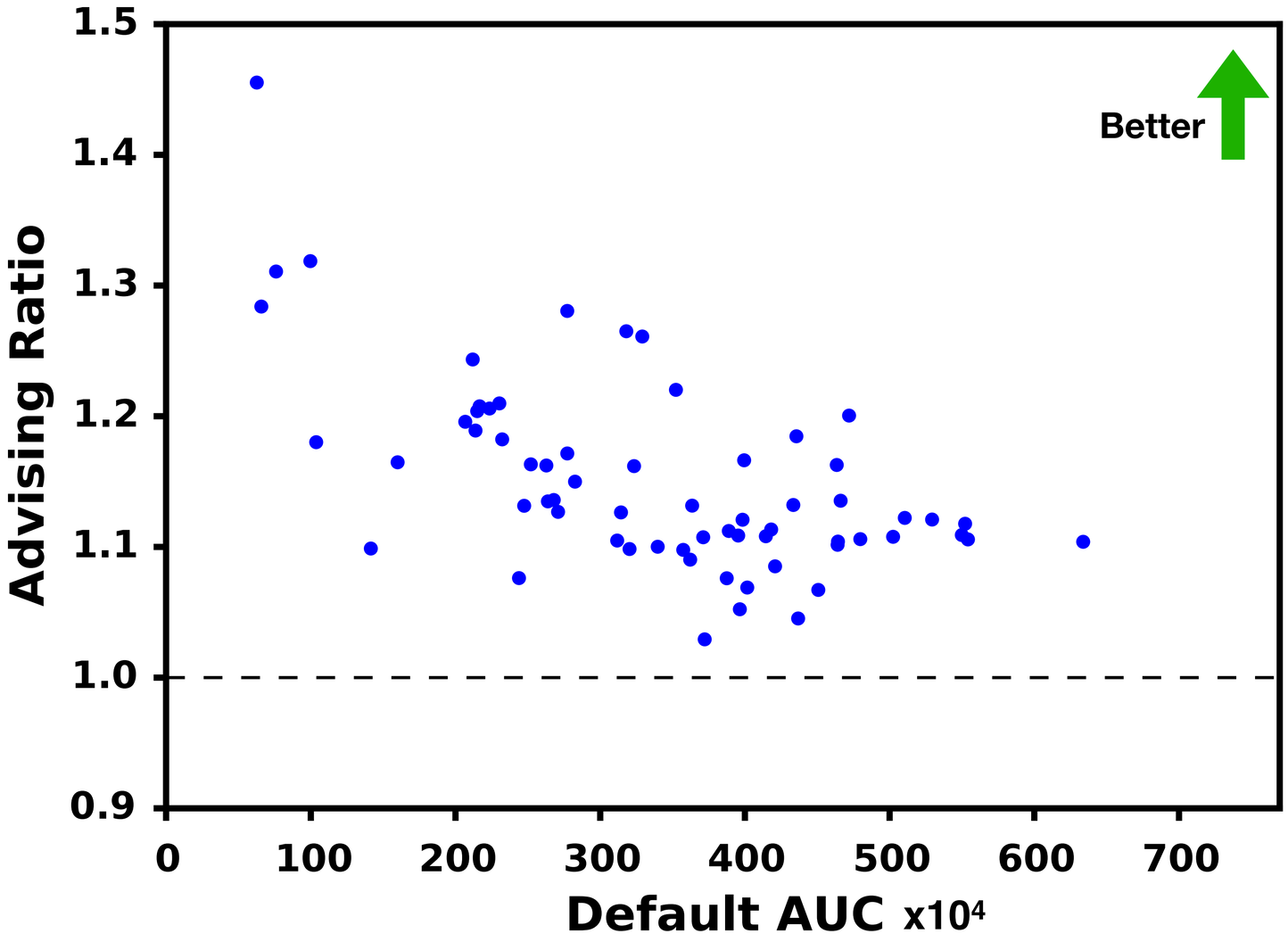
StringTie advising



11.1% average increase in accuracy for Coordinate Ascent

[Pertea, *et al.*, Nature Biotechnology 2015]

StringTie advising

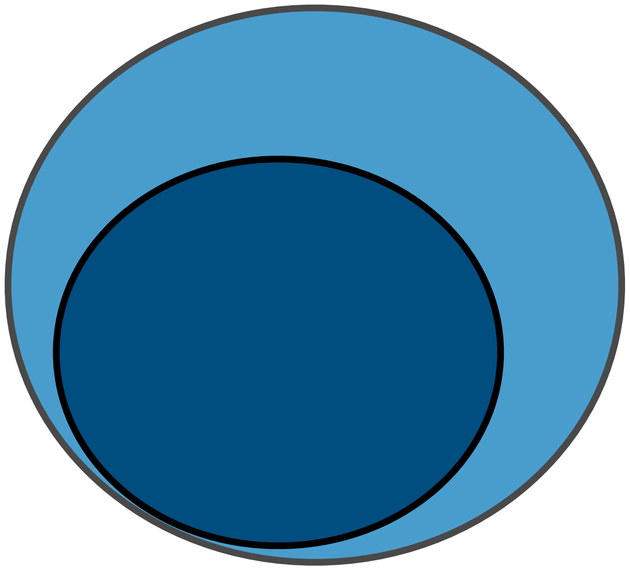


- all aligned RNA-seq from ENCODE
- variety of aligners
- example of performance in general

average advising ratio: 1.151

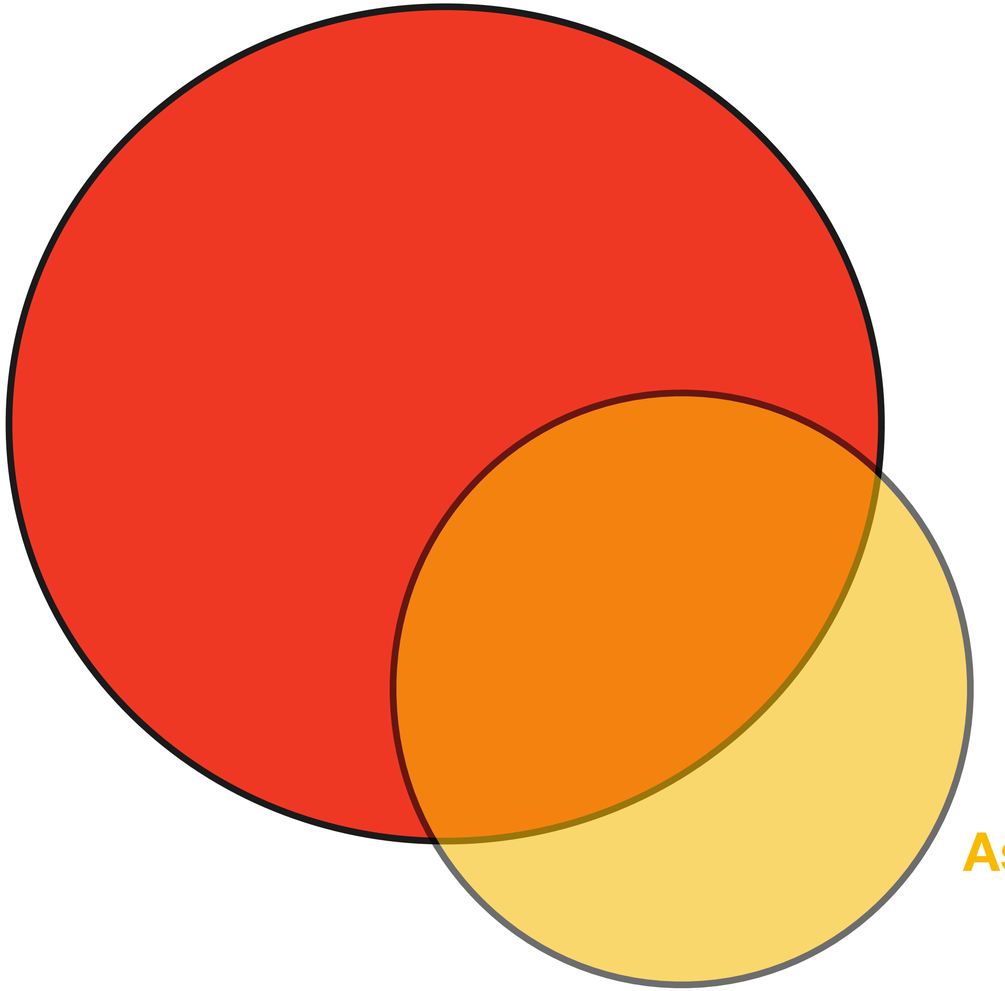
AUC vs other metrics

Sequencing Reads



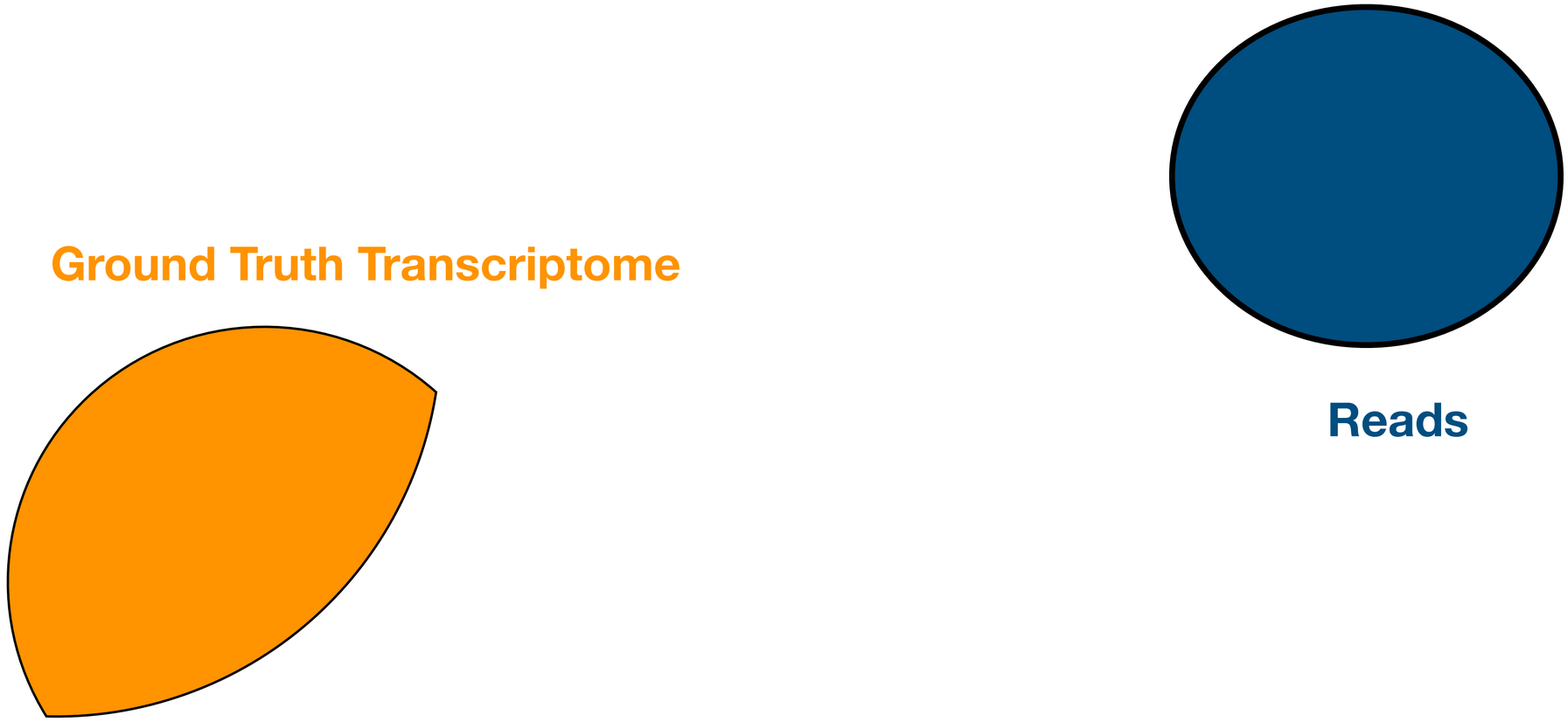
Mapped Reads

Reference Transcriptome

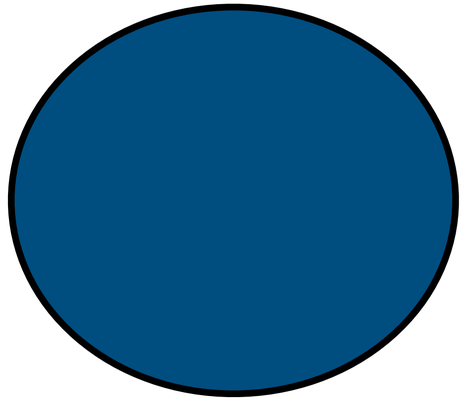


Assembled Transcriptome

AUC vs other metrics

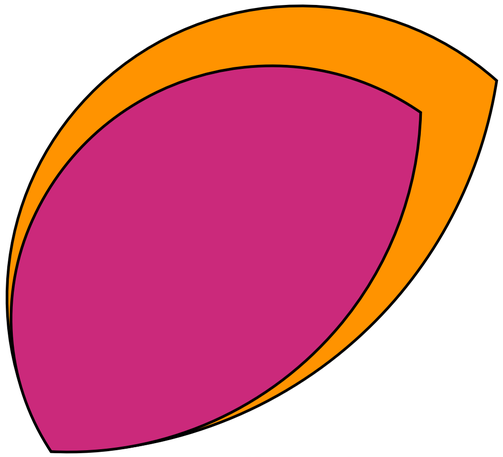


AUC vs other metrics



Reads

Ground Truth Transcriptome

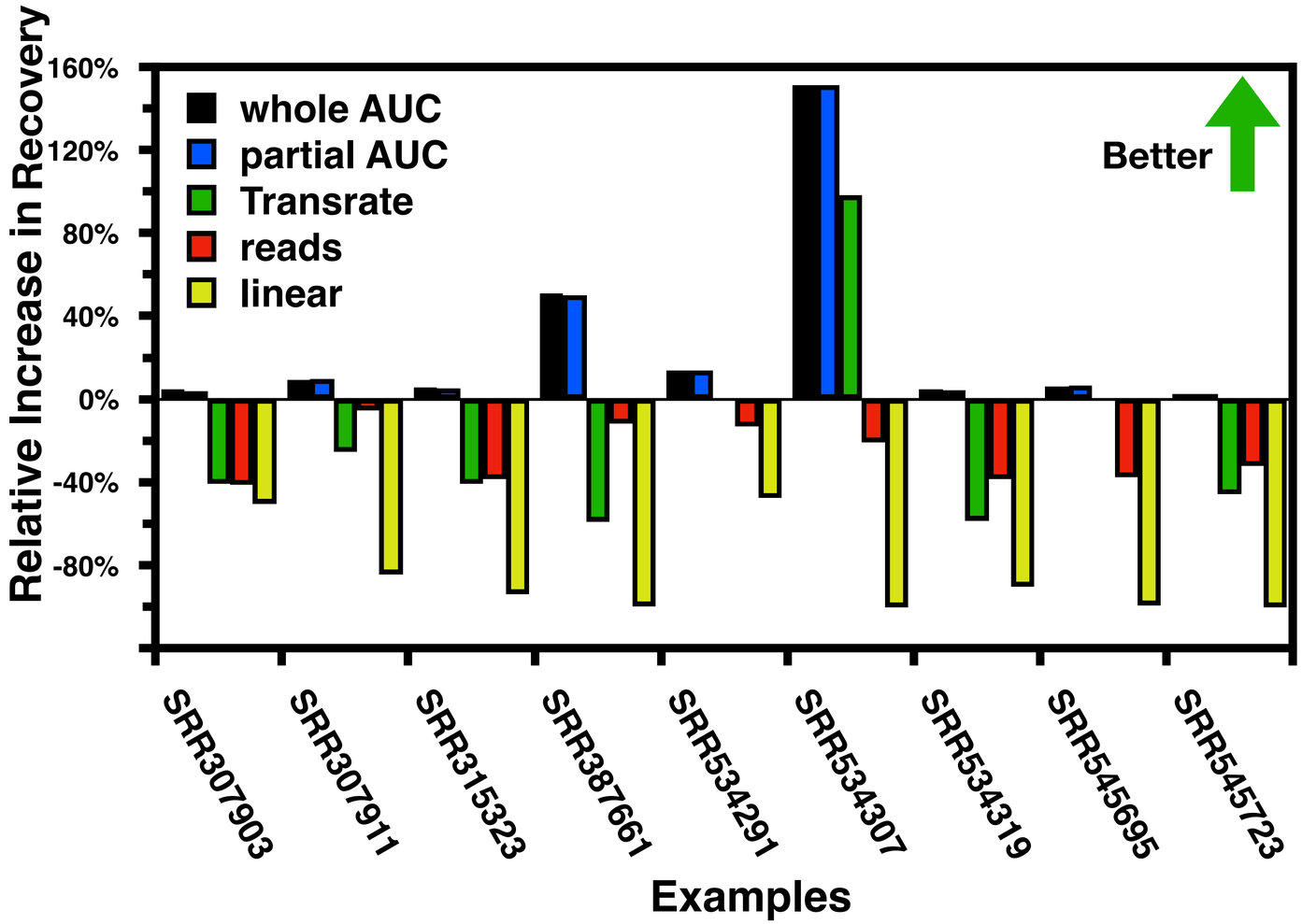


"Reference" Transcriptome

AUC vs other metrics

AUC penalizes all transcripts that don't map to the reference

- simulated data where we know the "novel" transcripts
- optimized using coordinate ascent and various metrics
- recovery rate of the reference & novelty compared to default



AUC is the only tested method to increase recovery when optimized

Summary

Parameter advising increases AUC for transcript assembly.

- Coordinate ascent is a novel method for advisor set construction.
- Advisor subsets can be used to reduce the resource requirements.
- Improvements are seen for both Scallop and StringTie.
- AUC is currently the best optimization metric.

Extensions

Taking inspiration from methods used previously

- Transcript-level advising
- Meta-assembly

Acknowledgments

Kingsford Group

Especially:

Mingfu Shao

Guillaume Marçais

Heewook Lee

Minh Hoang

Published at the **Workshop on Computational Biology at ICML 2019**

Slides (and links): **dandeblasio.com/AutoAlg19**

Scallop Advising: https://github.com/Kingsford-Group/scallopadvising

Funding

Gordon and Betty Moore Foundation's Data-Driven Discovery Initiative (GBMF4554)

US National Science Foundation (CCF-1256087 and CCF-1319998)

US National Institutes of Health (R01HG007104 and R01GM122935)

The Shurl and Kay Curci Foundation

The University of Texas at El Paso

Computational Biology Department

Carnegie Mellon University
School of Computer Science

GORDON AND BETTY MOORE FOUNDATION