Toward building an automated bioinformatician: parameter advising for improved scientific discovery

Dan DeBlasio
dandeblasio.com
danfdeblasio

slides: www.dandeblasio.com/automated-bioinformatician/

Computational Biology Department

Carnegie Mellon University
School of Computer Science

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 problems in bioinformatics

Going to focus on two building blocks of bioinformatics analysis

Transcript assembly

- 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.

Multiple sequence alignment

- Matches regions of similarity between sequences.
- Used to construct trees, search for differences.

Transcript assembly (TA)

Given

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

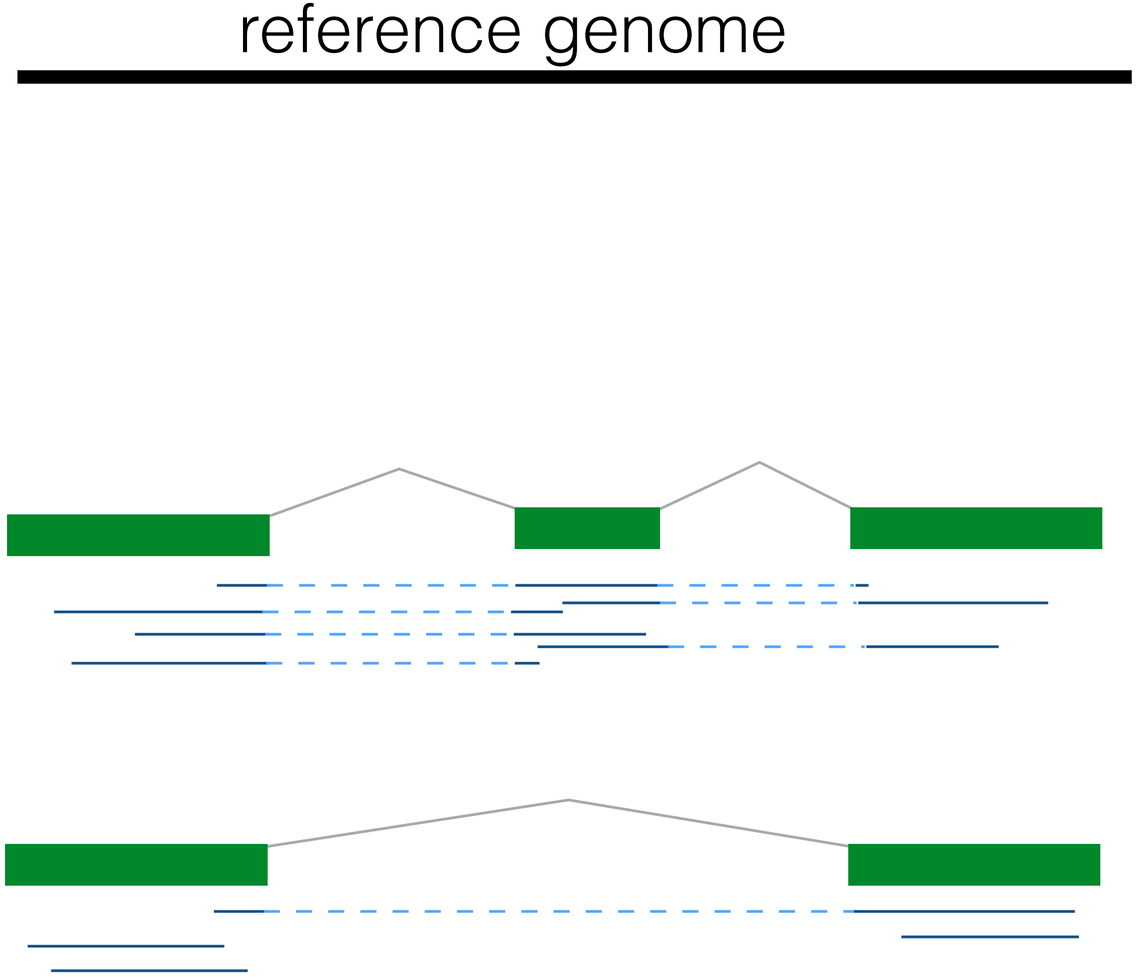
Transcript assembly (TA)

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.



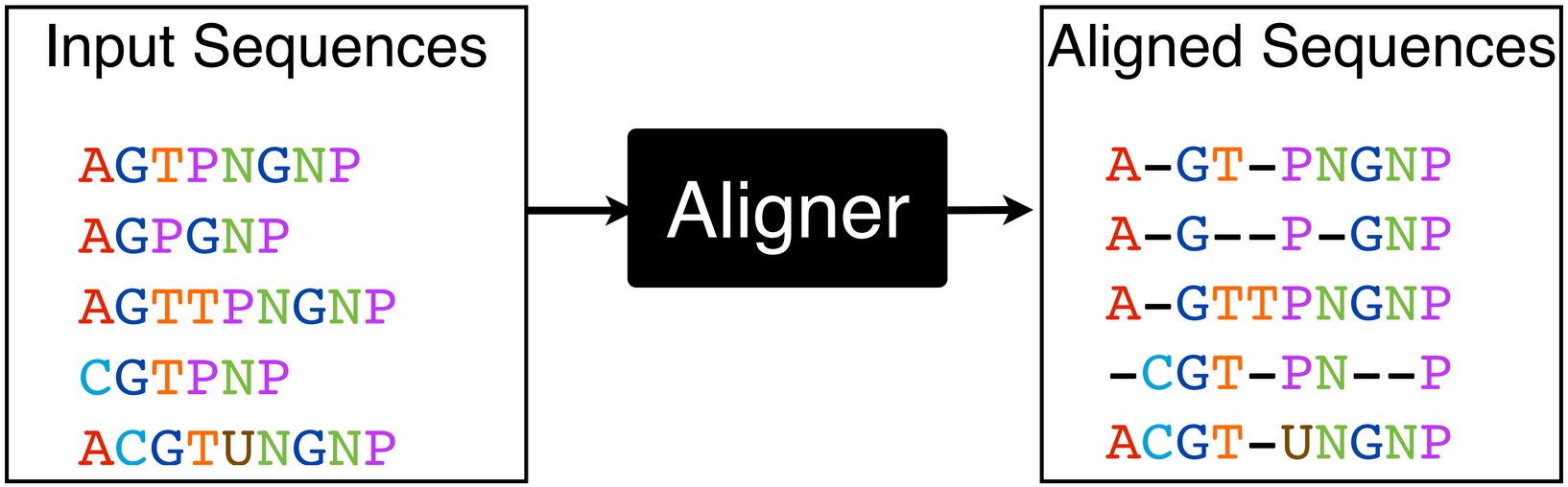
Multiple sequence alignment (MSA)

Given

- a set of sequences $\{S_1, S_2, ..., S_m\}$, and
- an alignment objective function

find an m x n matrix

- where each row represents one sequence from the set with inserted gaps, and
- is optimal under the objective function.



Bioinformatics software

TA and MSA, like many others, are fundamental problems in bioinformatics.

- 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

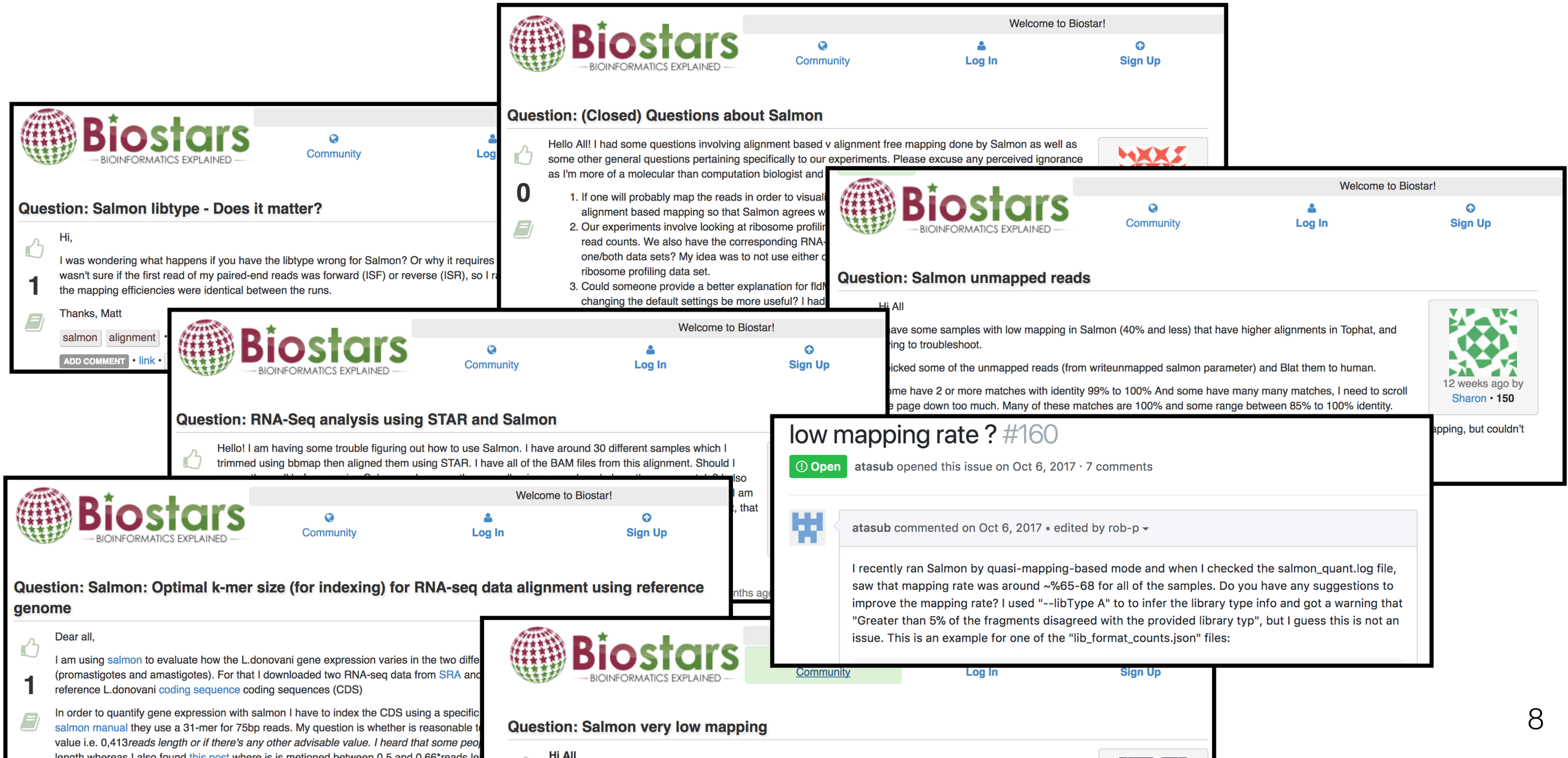
```
Quant
==========
Perform dual-phase, mapping-based estimation of
transcript abundance from RNA-seq reads

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
```

--reduceGCMemory	If this option is selected, a more memory efficient (but slightly slower) representation is used to compute fragment GC content. Enabling this will reduce memory usage, but can also reduce speed. However, the results themselves will remain the same.
--biasSpeedSamp arg	The value at which the fragment length PMF is down-sampled when evaluating sequence-specific & GC fragment bias. Larger values speed up effective length correction, but may decrease the fidelity of bias modeling results.
--strictIntersect	Modifies how orphans are assigned. When this flag is set, if the intersection of the quasi-mappings for the left and right is empty, then all mappings for the left and all mappings for the right read are reported as orphaned quasi-mappings
--fldMax arg	The maximum fragment length to consider when building the empirical distribution
--fldMean arg	The mean used in the fragment length distribution prior
--fldSD arg	The standard deviation used in the fragment length distribution prior
-f [--forgettingFactor] arg	The forgetting factor used in the online learning schedule. A smaller value results in quicker learning, but higher variance and may be unstable. A larger value results in slower learning but may be more stable. Value should be in the interval (0.5, 1.0].
-m [--maxOcc] arg	(S)MEMs occuring more than this many times won't be considered.
--initUniform	initialize the offline inference with uniform parameters, rather than seeding with online parameters.
-w [--maxReadOcc] arg	Reads "mapping" to more than this many places won't be considered.
--noLengthCorrection	[experimental] : Entirely disables length correction when estimating the abundance of transcripts. This option can be used with protocols where one expects that fragments derive from their underlying targets without regard to that target's length (e.g. QuantSeq)
--noEffectiveLengthCorrection	Disables effective length correction when computing the probability that a fragment was generated from a transcript. If this flag is passed in, the fragment length distribution is not taken into account when computing this probability.
--noFragLengthDist	[experimental] : Don't consider concordance with the learned fragment length distribution when trying to determine the probability that a fragment has originated from a specified location. Normally, Fragments with unlikely lengths will be assigned a smaller relative probability than those with more likely lengths. When this flag is passed in, the observed fragment length has no effect on that fragment's a priori probability.
--noBiasLengthThreshold	[experimental] : If this option is enabled, then no (lower) threshold will be set on how short bias correction can make effective lengths. This can increase the precision of bias correction, but harm robustness. The default correction applies a threshold.
--numBiasSamples arg	Number of fragment mappings to use when learning the sequence-specific bias model.
--numAuxModelSamples arg	The first <numAuxModelSamples> are used to train the auxiliary model parameters (e.g. fragment length distribution, bias, etc.). After ther first <numAuxModelSamples> observations the auxiliary model parameters will be assumed to have converged and will be fixed.
--numPreAuxModelSamples arg	The first <numPreAuxModelSamples> will have their assignment likelihoods and contributions to the transcript abundances computed without applying any auxiliary models. The purpose of ignoring the auxiliary models for the first <numPreAuxModelSamples> observations is to avoid applying these models before thier parameters have been learned sufficiently well.
--useVBOpt	Use the Variational Bayesian EM rather than the traditional EM algorithm for optimization in the batch passes.
--rangeFactorizationBins arg	Factorizes the likelihood used in quantification by adopting a new notion of equivalence classes based on the conditional probabilities with which fragments are generated from different transcripts. This is a more fine-grained factorization than the normal rich equivalence classes. The default value (0) corresponds to the standard rich equivalence classes, and larger values imply a more fine-grained factorization. If range factorization is enabled, a common value to select for this parameter is 4.
--numGibbsSamples arg	Number of Gibbs sampling rounds to perform.
--numBootstraps arg	Number of bootstrap samples to generate. Note: This is mutually exclusive with Gibbs sampling.
--thinningFactor arg	Number of steps to discard for every sample kept from the Gibbs chain. The larger this number, the less chance that subsequent samples are auto-correlated, but the slower sampling becomes.
-q [--quiet]	Be quiet while doing quantification (don't write informative output to the console unless something goes wrong).
--perTranscriptPrior	The prior (either the default or the argument provided via --vbPrior) will be interpreted as a transcript-level prior (i.e. each transcript will be given a prior read count of this value)
--vbPrior arg	The prior that will be used in the VBEM algorithm. This is interpreted as a per-nucleotide prior, unless the --perTranscriptPrior flag is also given, in which case this is used as a transcript-level prior
--writeOrphanLinks	Write the transcripts that are linked by orphaned reads.
--writeUnmappedNames	Write the names of un-mapped reads to the file unmapped_names.txt in the auxiliary directory.
-x [--quasiCoverage] arg	[Experimental]: The fraction of the read that must be covered by MMPs (of length >= 31) if this read is to be considered as "mapped". This may help to avoid "spurious" mappings. A value of 0 (the default) denotes no coverage threshold (a single 31-mer can yield a mapping). Since coverage by exact matching, large, MMPs is a rather strict condition, this value should likely be set to something low, if used.

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

```
... yl-lhqflspssnqrtdqyggsvenrarlvlevvdavcnewsad-RIGIRVSPigtfq ...
... kP-LGVKLPPyf--dlvhfdimaeilnqfpltyvsnv-nsig----nglfidpeaesv ...
... yl-lnqfldphsntrtdeyggsienrarftlevvdalveaighe-KVGLRLSPygvfn ...
... yl-plqflnpyynkrtdkyggslenrarfwletlekvkhavgsdcAIATRF---GVdt ...
... kvPLYVKLSPnv-tdivpiakaveaagadgltmintl---------mgvrfdlktrqp ...
```

alternate

```
... gsvenrarlvlevvdavcnewsad-RIGIRVSPigtfqnvdngpnee--adalyl--- ...
... ydfeatekllke-----vftfftk-PLGVKLPPyf---------------dlvhfdim ...
... gsienrarftlevvdalveaighe-KVGLRLSPygvfnsmsggaetgivaqyayvage ...
... gslenrarfwletlekvkhavgsdcAIATRFGV-----------------dtvygpgq ...
... tdpevaaalvka-----ckavskv-PLYVKLSPnvt--------------divpiaka ...
```

The default parameter choices misaligns this region of the sequences.

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

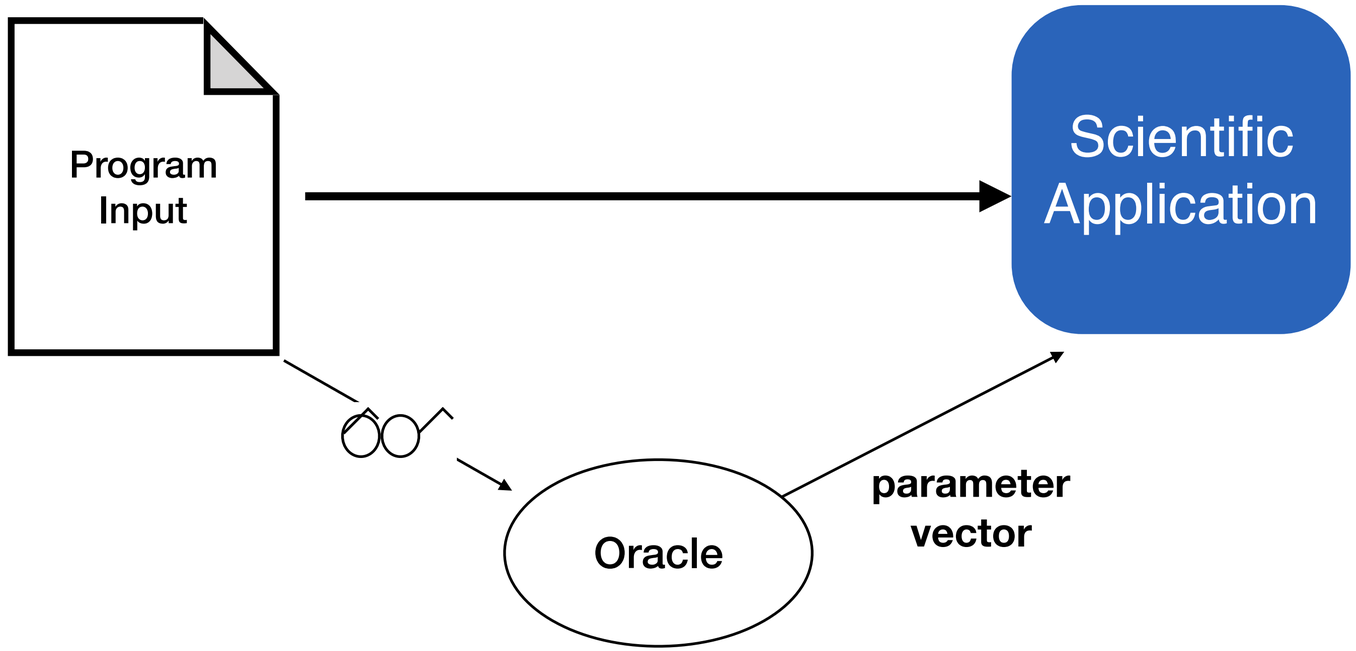
Almost all pieces of scientific software have tunable parameters.

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

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

Automated bioinformatician

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



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.

A posteriori advising

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

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

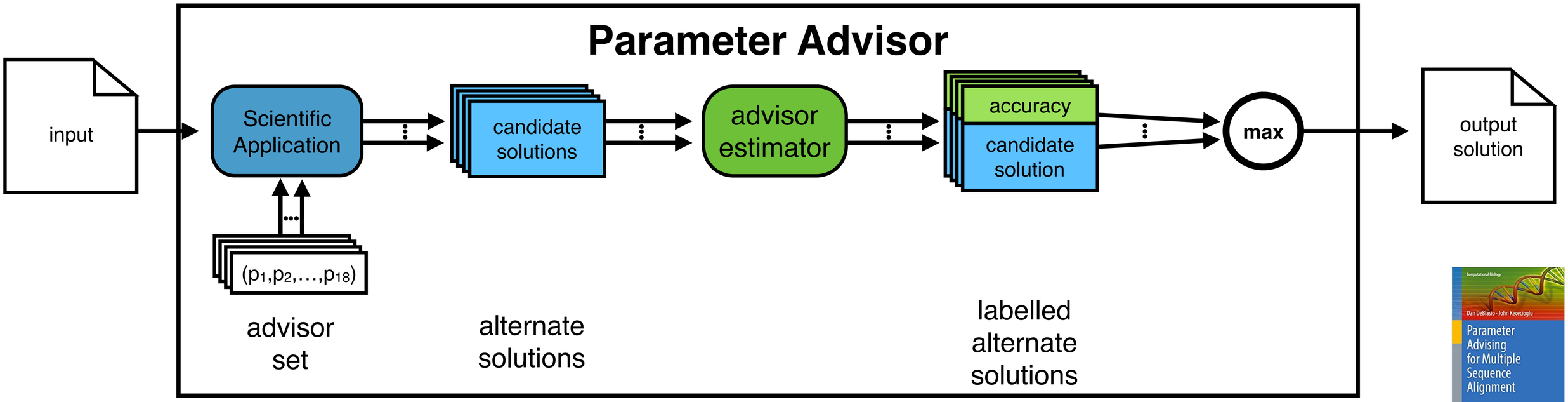
Issue is that running time is increased greatly.

- 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.



[DK, Springer International, 2017]

Parameter advising framework

Steps of advising:

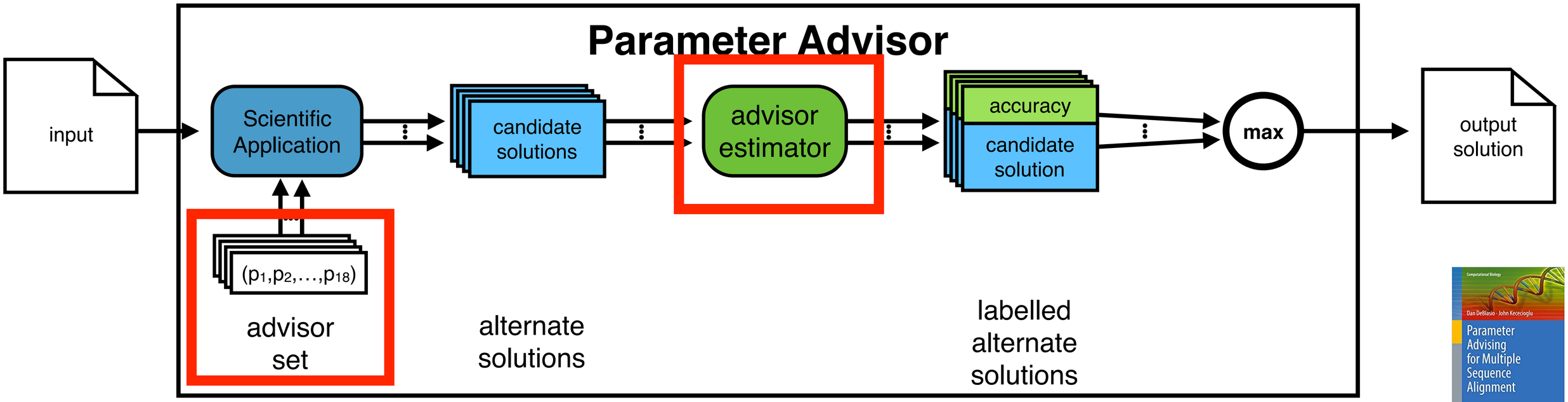
- An advisor set of parameter choice vectors is used to obtain candidates.
- Solutions are ranked based on the accuracy estimation.
- The highest ranked candidate is returned.

[DK, Springer International, 2017]

Parameter advising framework

Components of an advisor:

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



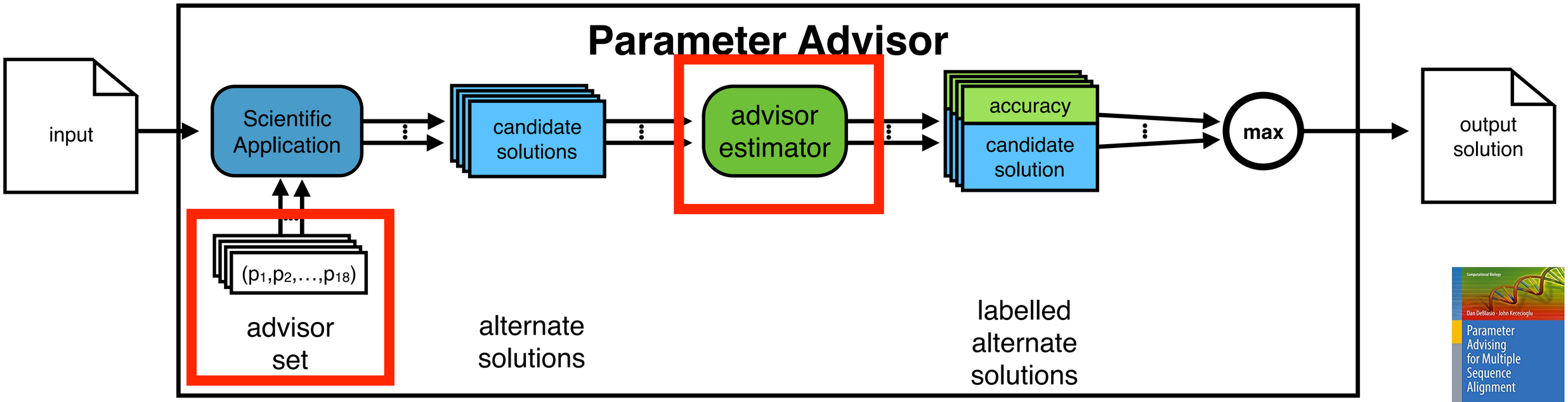
[DK, Springer International, 2017]

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



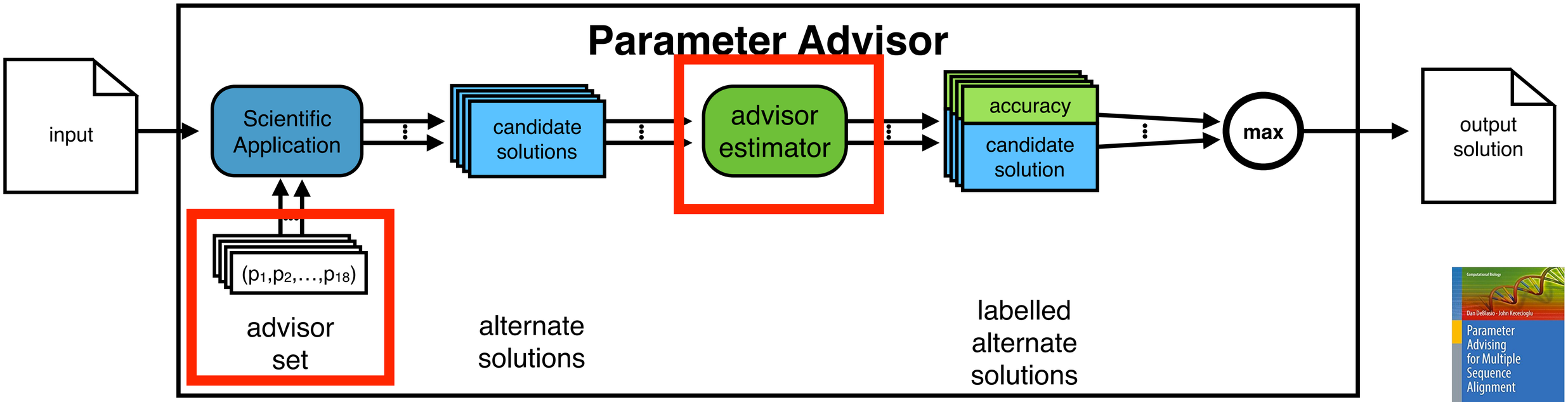
[DK, Springer International, 2017]

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



[DK, Springer International, 2017]

Today's topics

➡ **Parameter advising in transcript assembly**

- Constructing advisor sets [DKK, bioRxiv, *under review*]

➡ **Parameter advising in multiple sequence alignment**

- Constructing an accuracy estimator [DWK RECOMB/JCB 2012, DK WABI/AMB 2016]
- Constructing advisor sets [DK ACM-BCB/TCBB 2014]

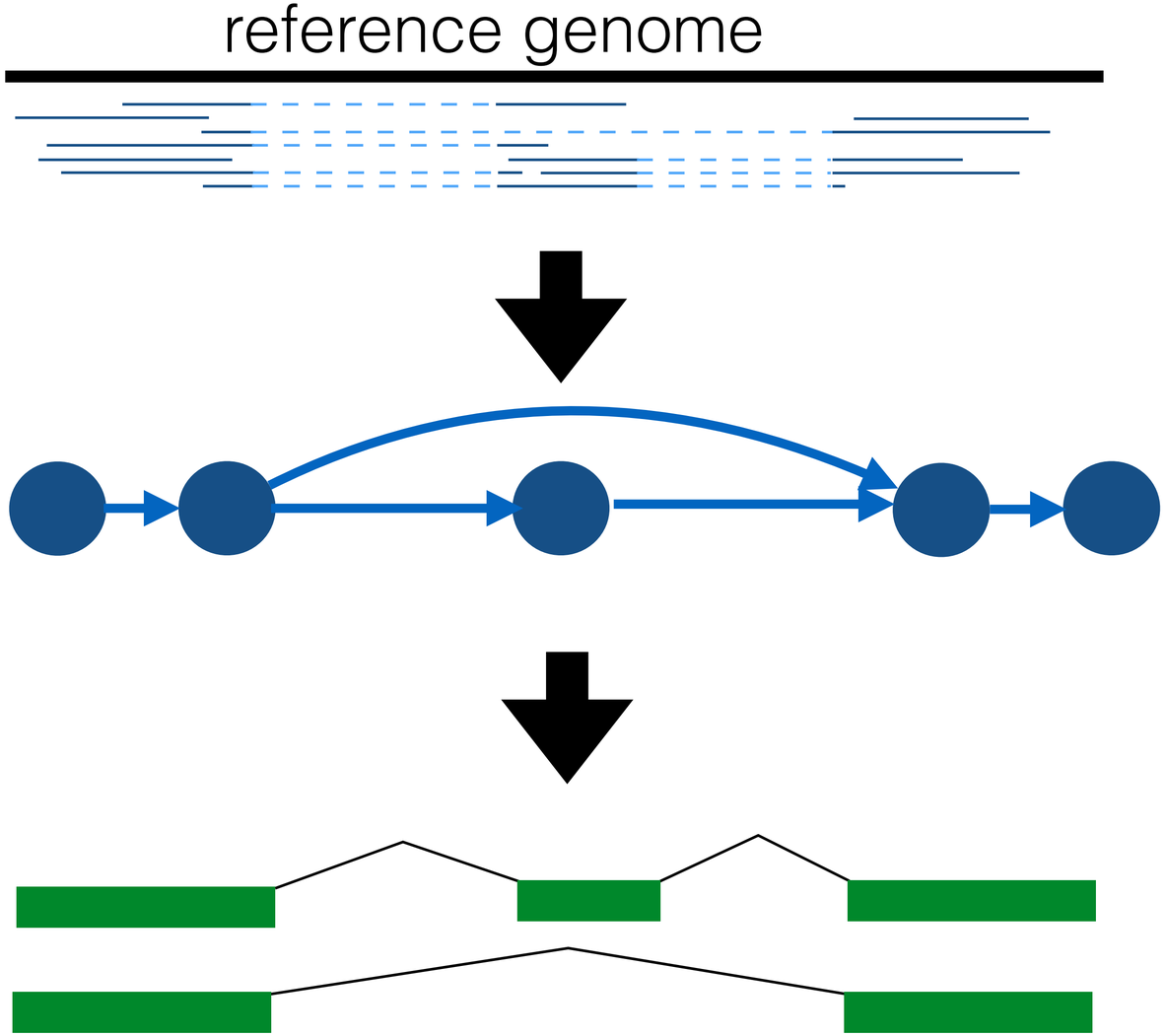
➡ **Extensions**

- Utilizing multiple applications [DK ACM-BCB 2015]
- Local improvement of solutions [DK RECOMB/JCB 2017]

Transcript assembly

TA is fundamental in transcriptomics.

- It's computationally difficult.
- It's easily impacted by choices of parameter values.
- There is no readily available way to confirm an assembly's accuracy.



[Shao and Kingsford, Nature Biotechnology 2017]

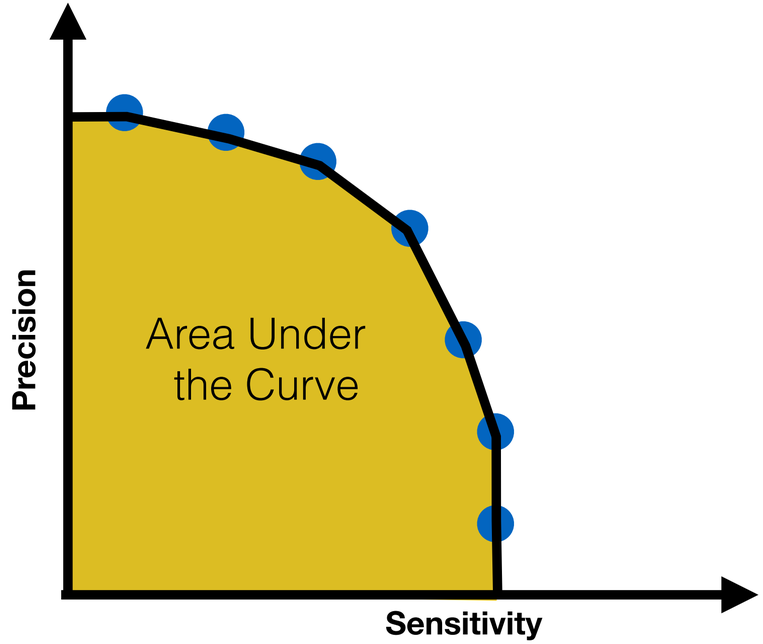
Transcript assembly

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.



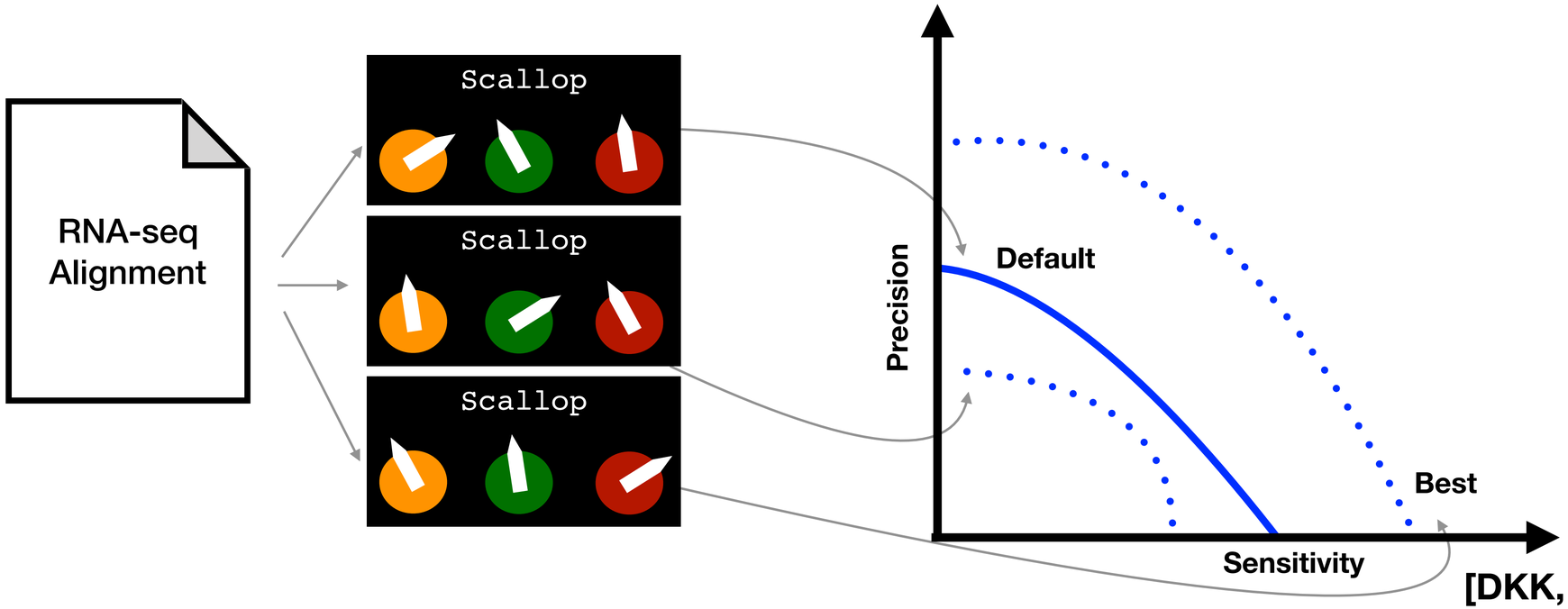
Transcript assembly advising

Advisor estimator:

- area under the curve

Advisor set:

- the number of tunable parameters is very large
- cannot exhaustively explore the space to find representative parameter vectors

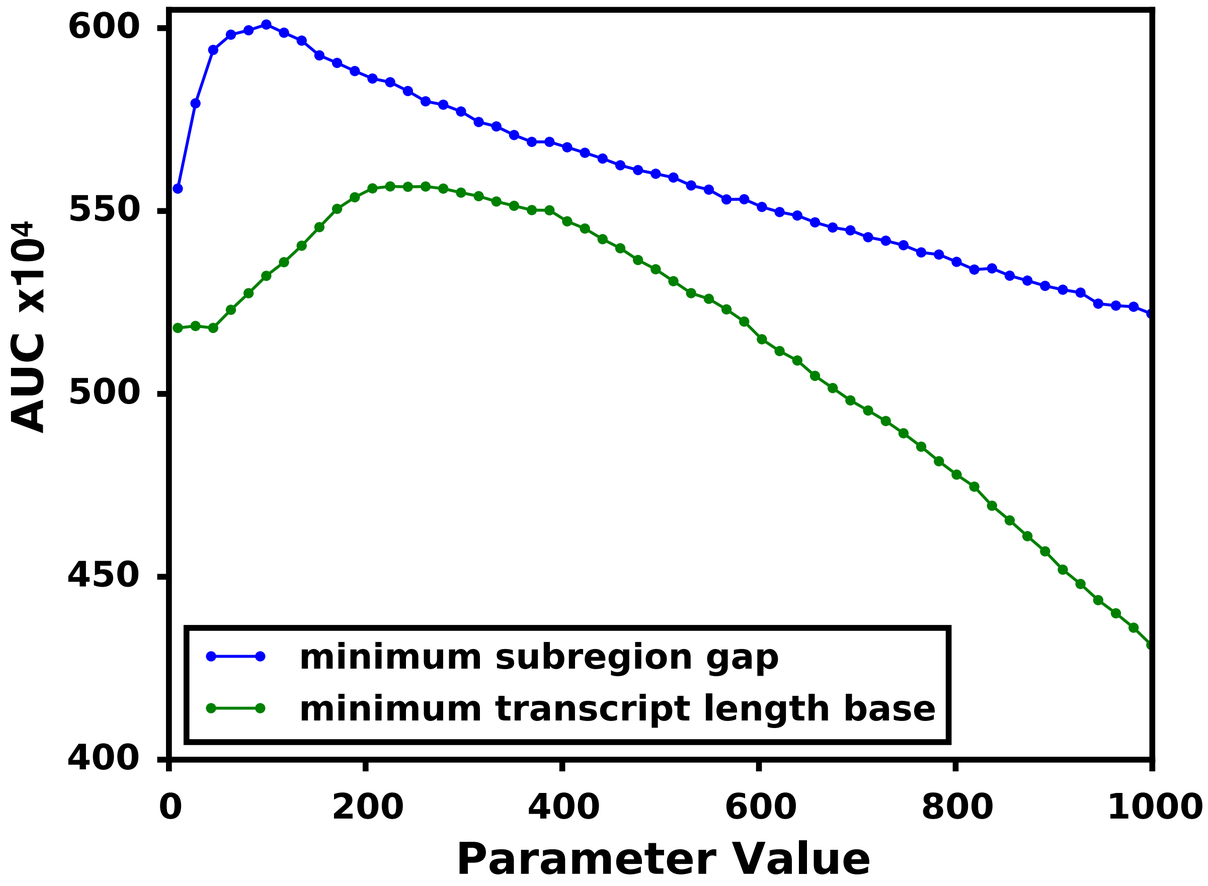


[DKK, bioRxiv, *under review*]

Finding an advisor set

Use information about parameter behavior to guide advisor set construction.

- Tested the influence of each parameter.
- Single maximum in the regions tested.

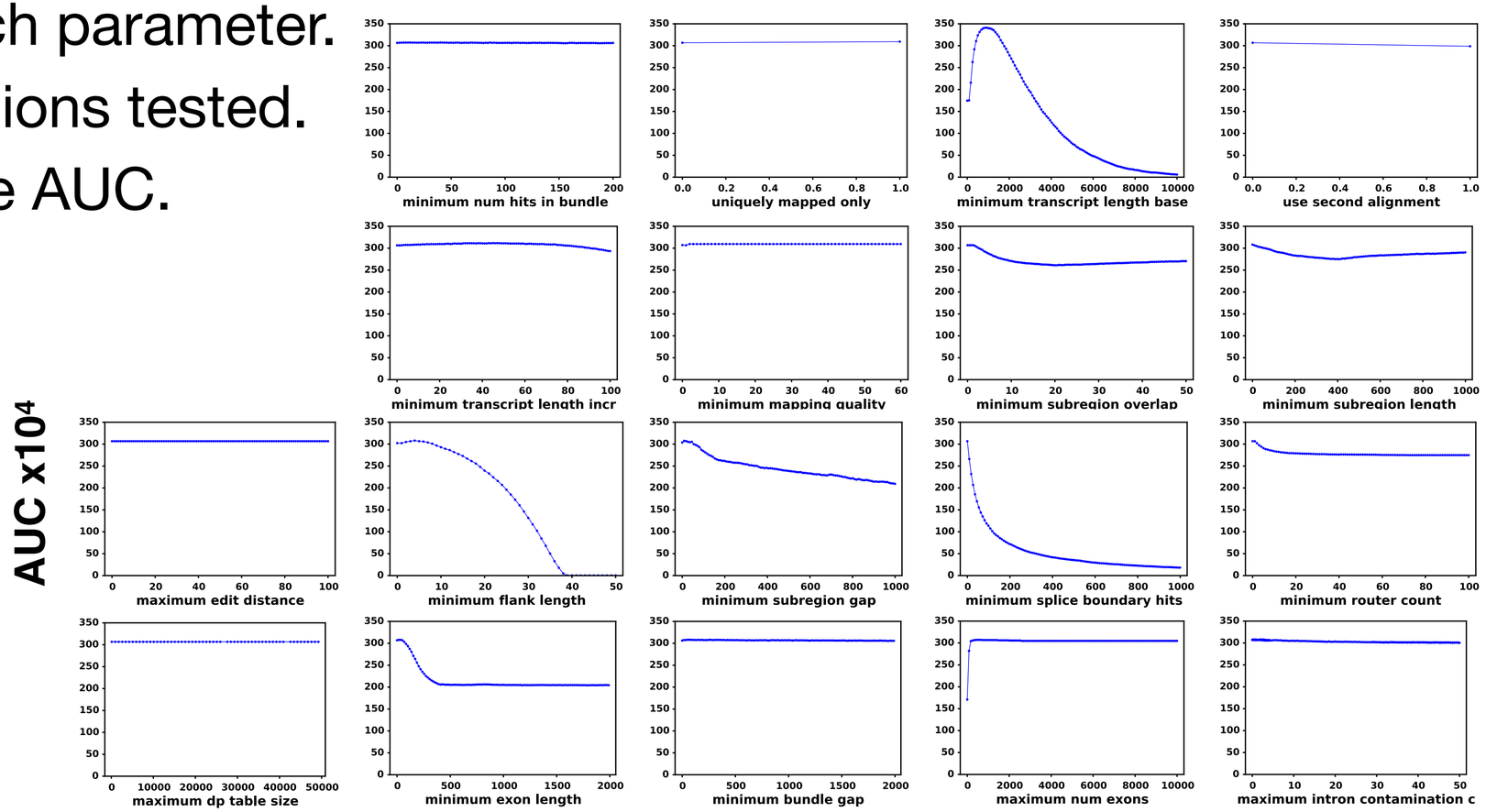


[DKK, bioRxiv, *under review*]

Finding an advisor set

Use information about parameter behavior to guide advisor set construction.

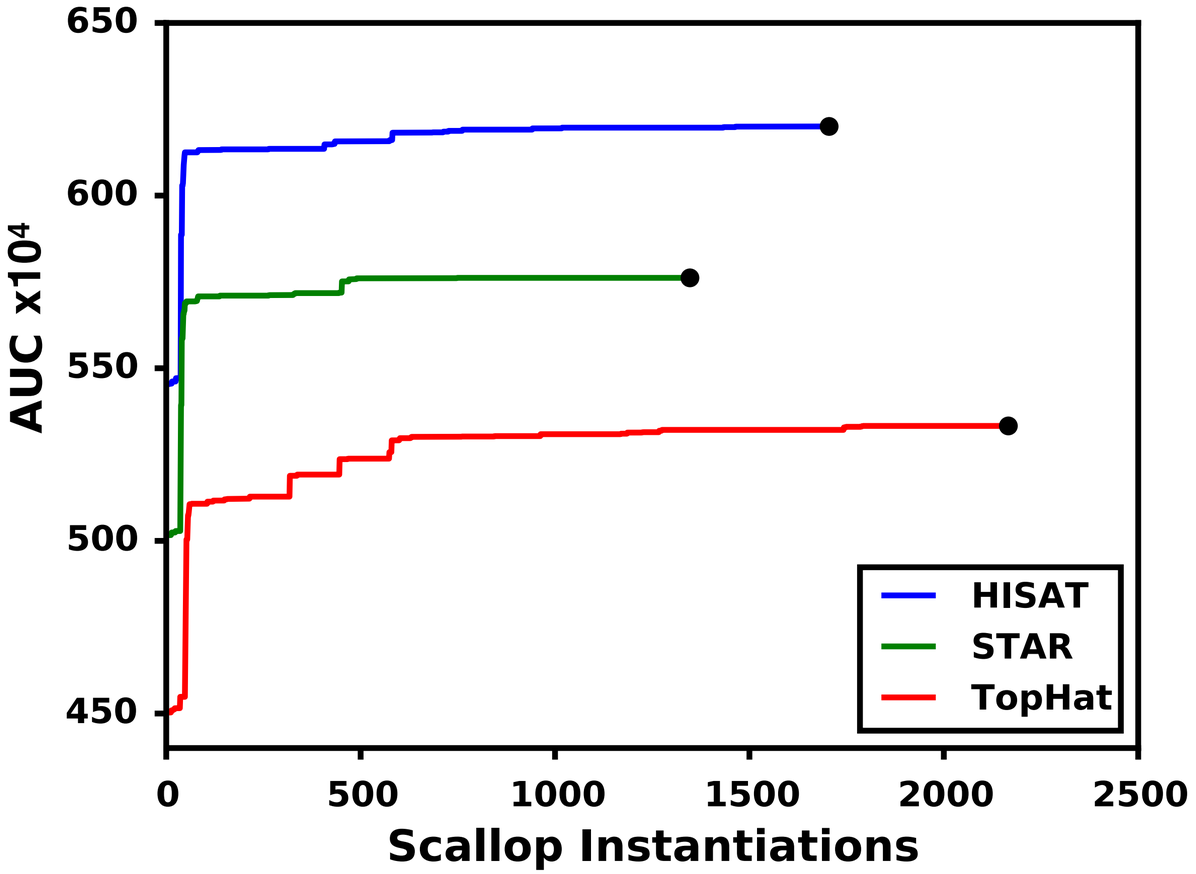
- Tested the influence of each parameter.
- Single maximum in the regions tested.
- Many parameters influence AUC.



Finding an advisor set

Parameter curve smoothness and single maxima help parameter selection.

- Iterative optimization will work well.
- Process is slow.



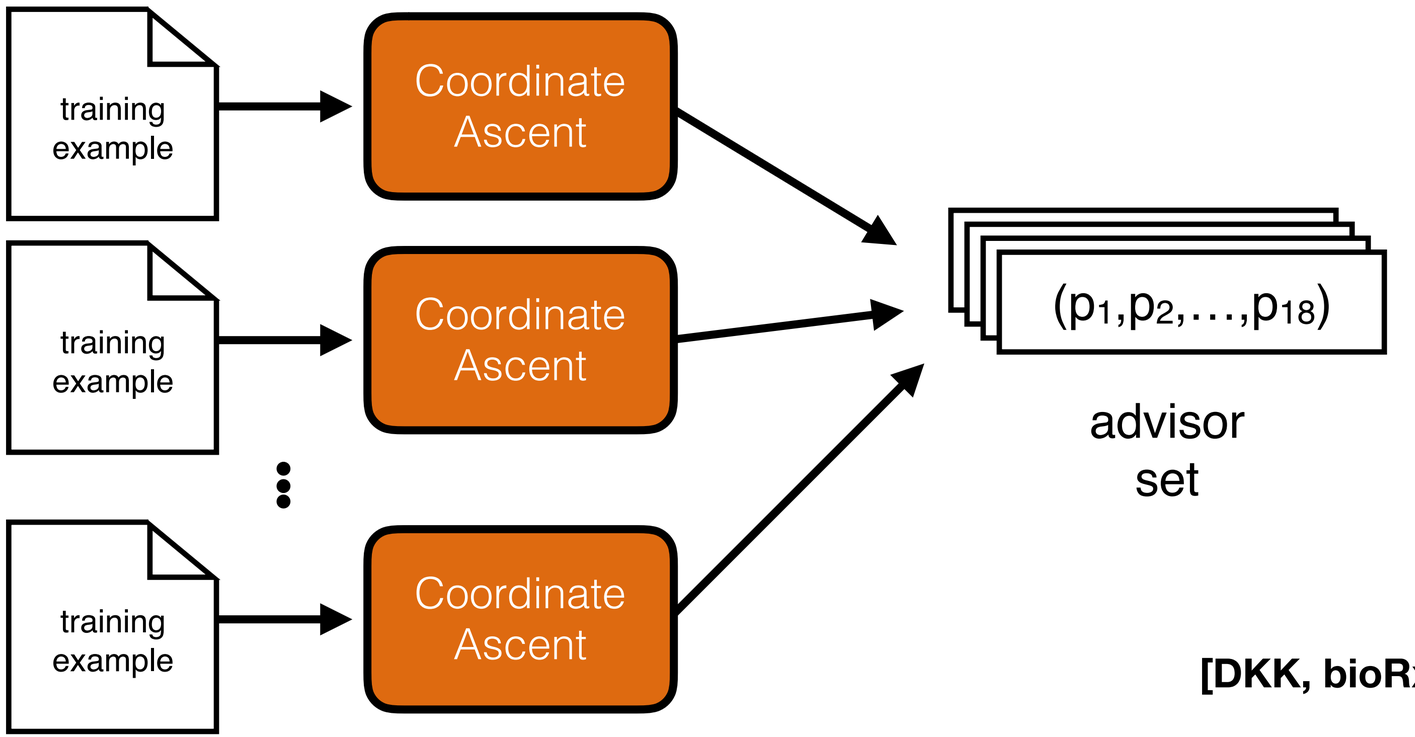
[DKK, bioRxiv, *under review*]

[HISAT: Kim, et al. Nat. Met. 2015] [STAR: Dobin , et al., Bioinformatics 2013] [TopHat: Kim, et al., Gen. Bio. 2013]

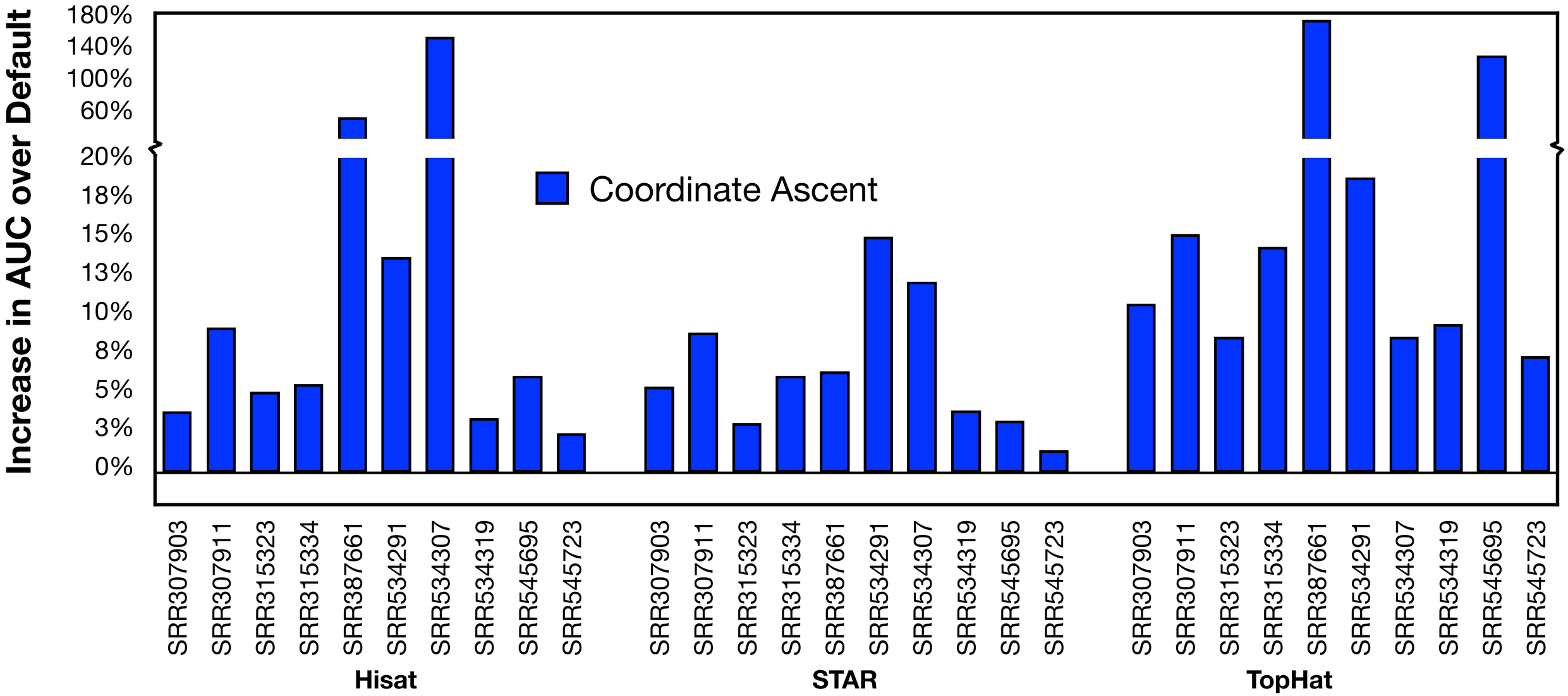
Finding an advisor set

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.



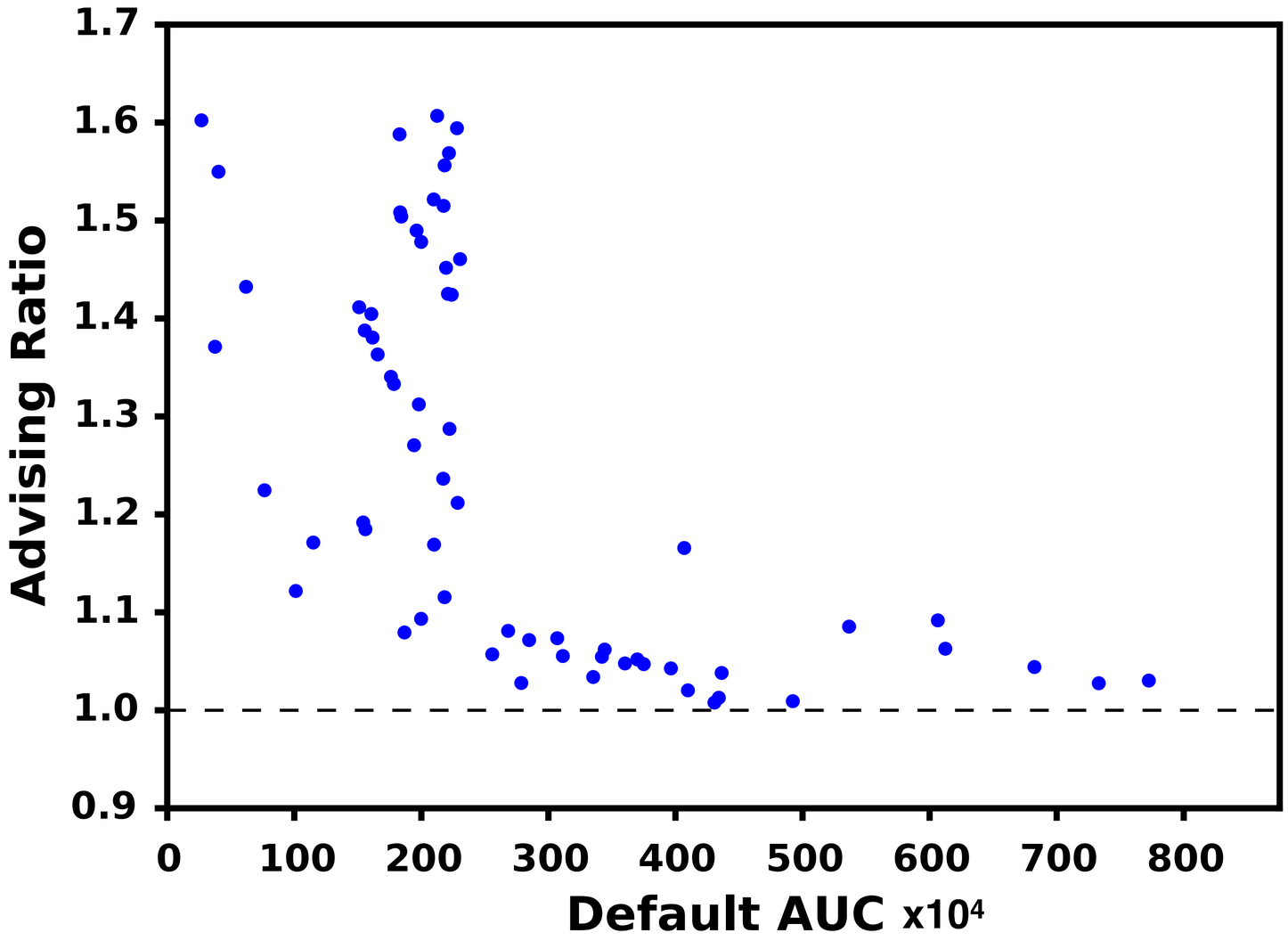
Scallop advising



Average of 18.1% increase in AUC using Coordinate Ascent

[DKK, bioRxiv, *under review*]

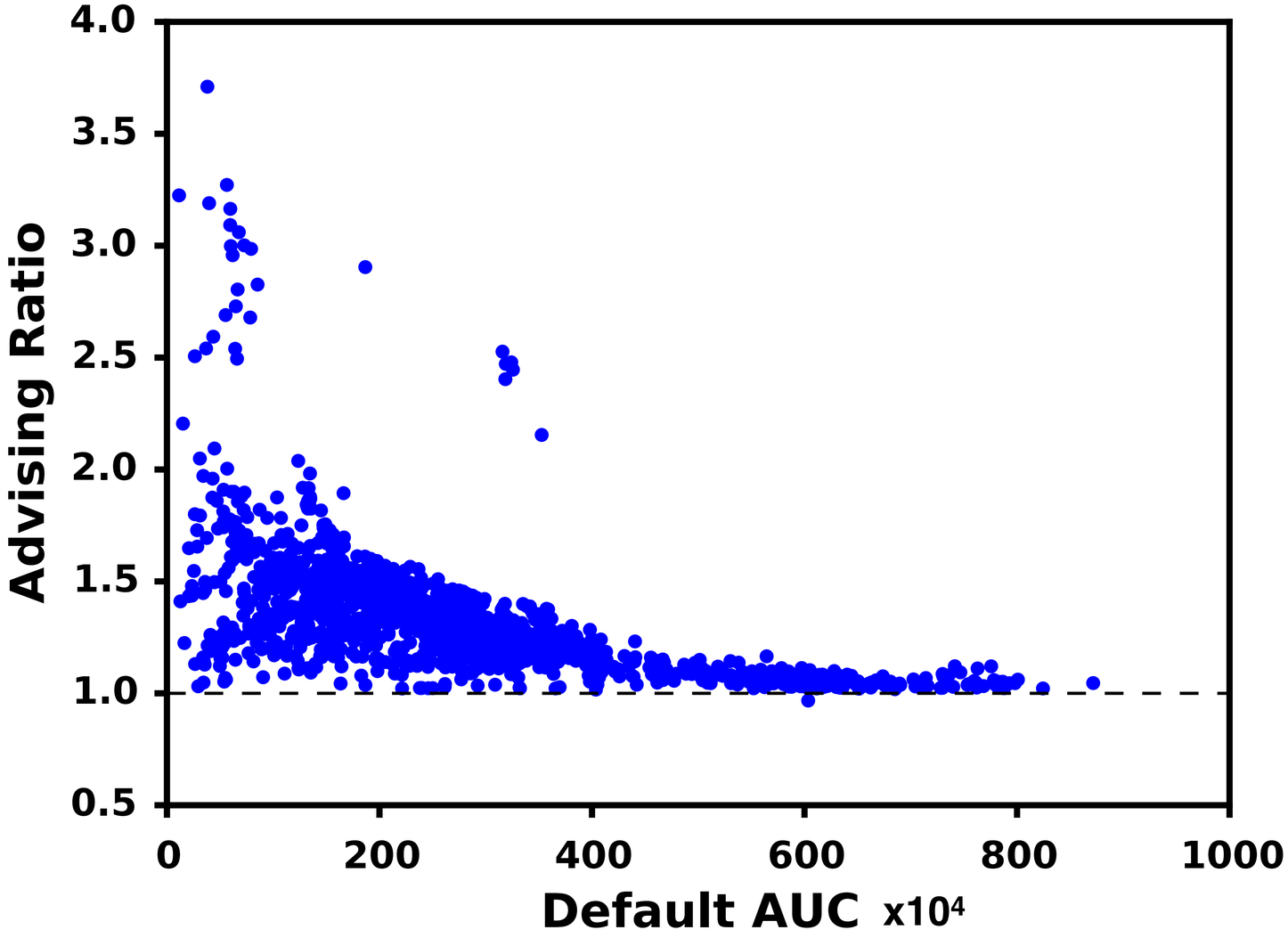
Scallop advising



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

average advising ratio: 1.257

Scallop advising

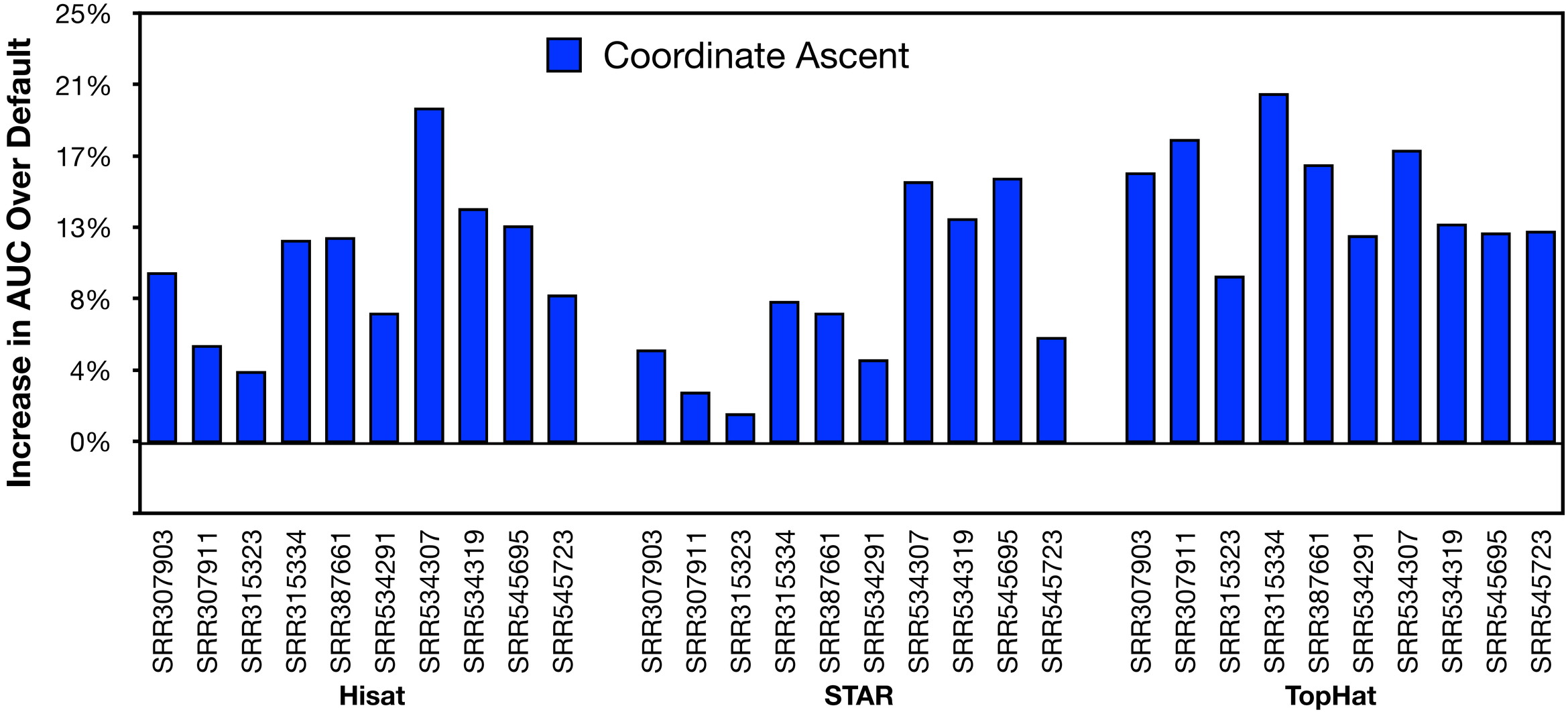


- 1595 RNA-Seq from SRA
- aligned using STAR
- example of high-throughput performance

average advising ratio: 1.382

[DKK, bioRxiv, *under review*]

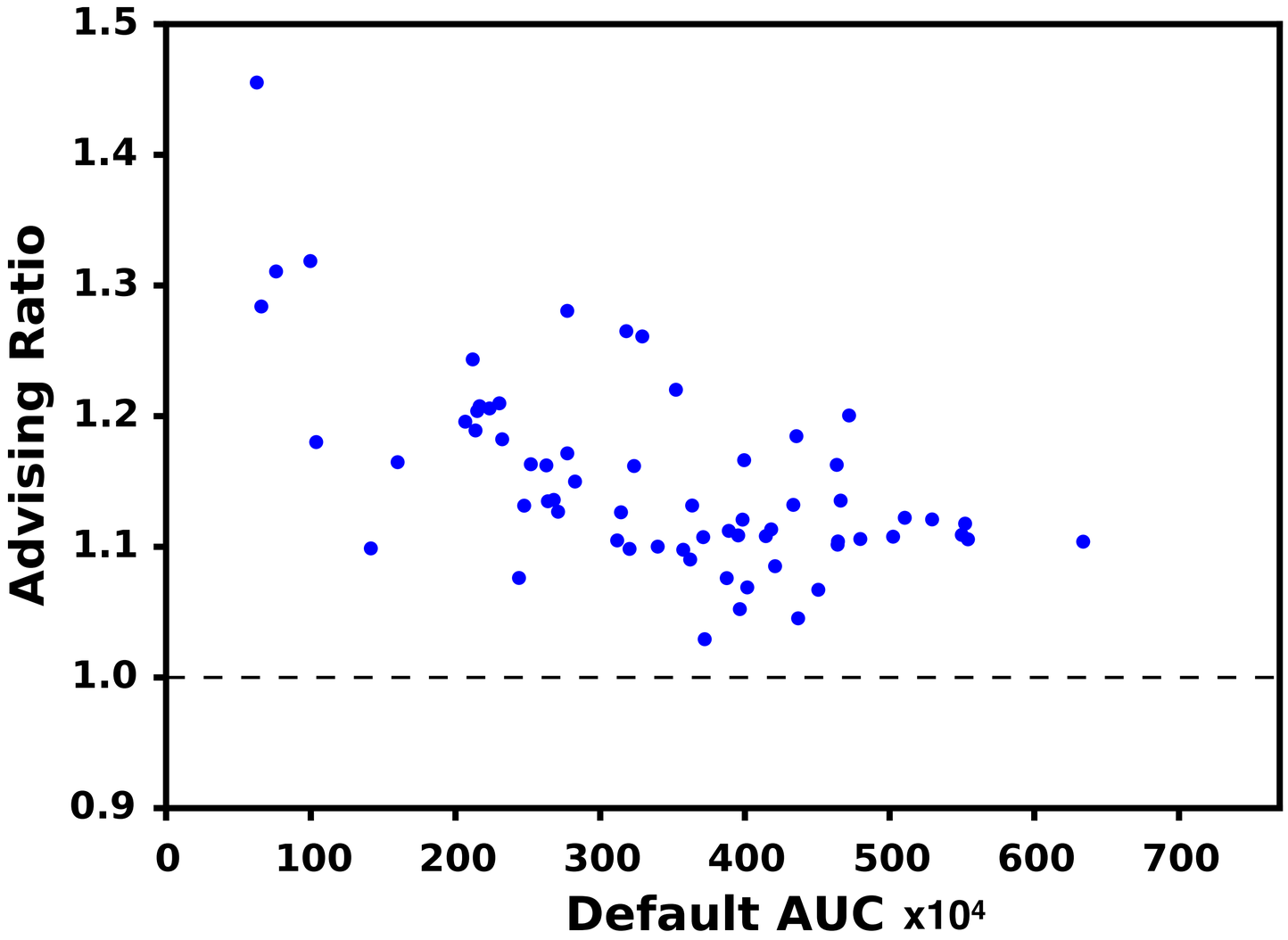
StringTie advising



11.1% average increase in accuracy for Coordinate Ascent

[DKK, bioRxiv, *under review*] [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

[DKK, bioRxiv, *under review*]

Transcript assembly advising

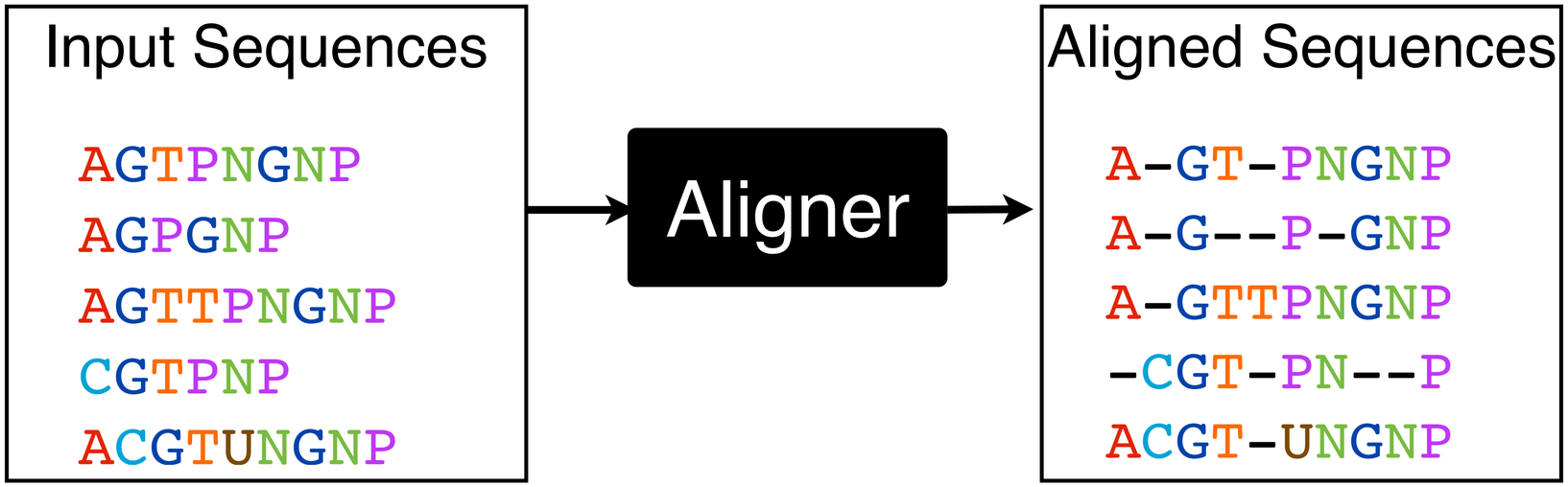
Parameter advising increases AUC for transcript assembly.

- Coordinate ascent is useful to advisor sets.
- Improvements are seen for both Scallop and StringTie.

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

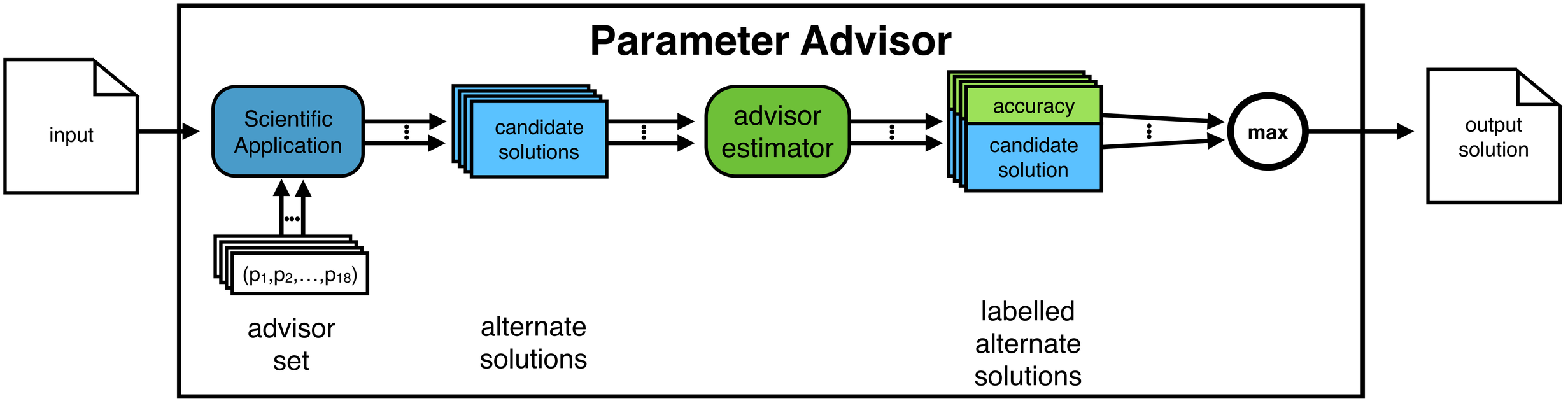


[Wheeler and Kececioglu ISMB 2007]

Parameter advising

Components of an advisor:

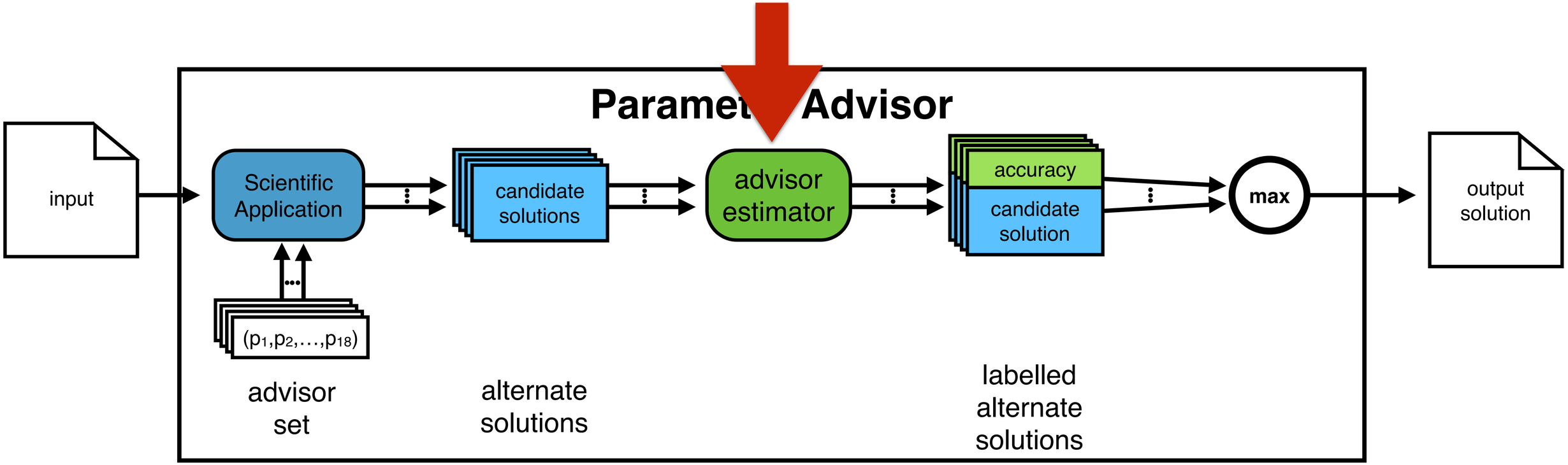
- An advisor set of parameter choice vectors.
- An advisor estimator to rank solutions.



Parameter advising

Components of an advisor:

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



Accuracy estimation

Alignment accuracy is measured with respect to a reference alignment.

reference alignment

```
… a D E h s …
… d S R - d …
… a S H l t …
```

computed alignment

```
… a D E h - s …
… d S R - - d …
… a S - H l t …
```

- accuracy is the fraction of substitutions from the reference that are in the computed alignment,

Accuracy estimation

Alignment accuracy is measured with respect to a reference alignment.

reference alignment

```
… a D E h s …
… d S R - d …
… a S H l t …
```

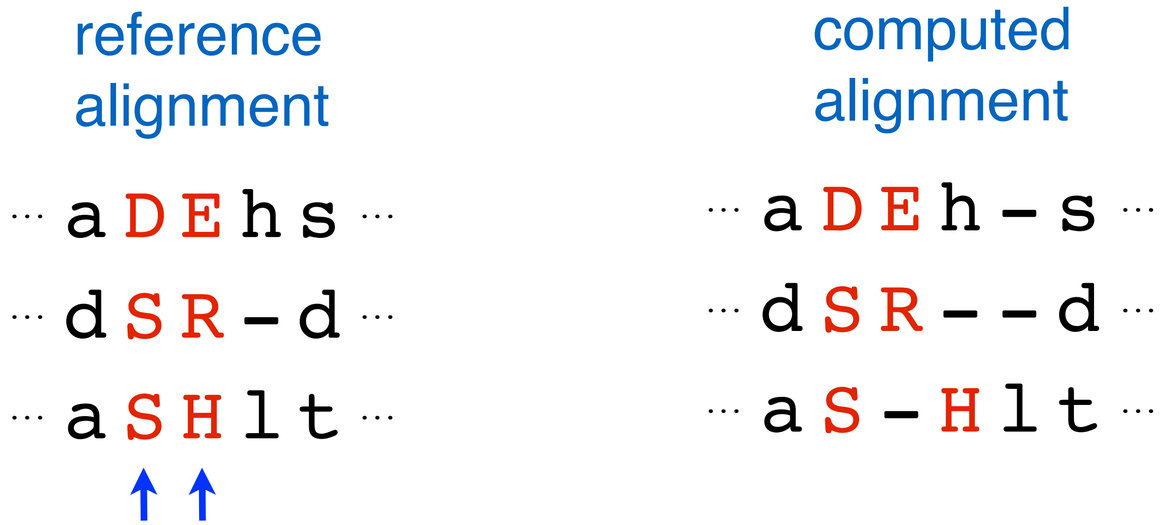
computed alignment

```
… a D E h - s …
… d S R - - d …
… a S - H l t …
```

- accuracy is the fraction of substitutions from the reference that are in the computed alignment,

Accuracy estimation

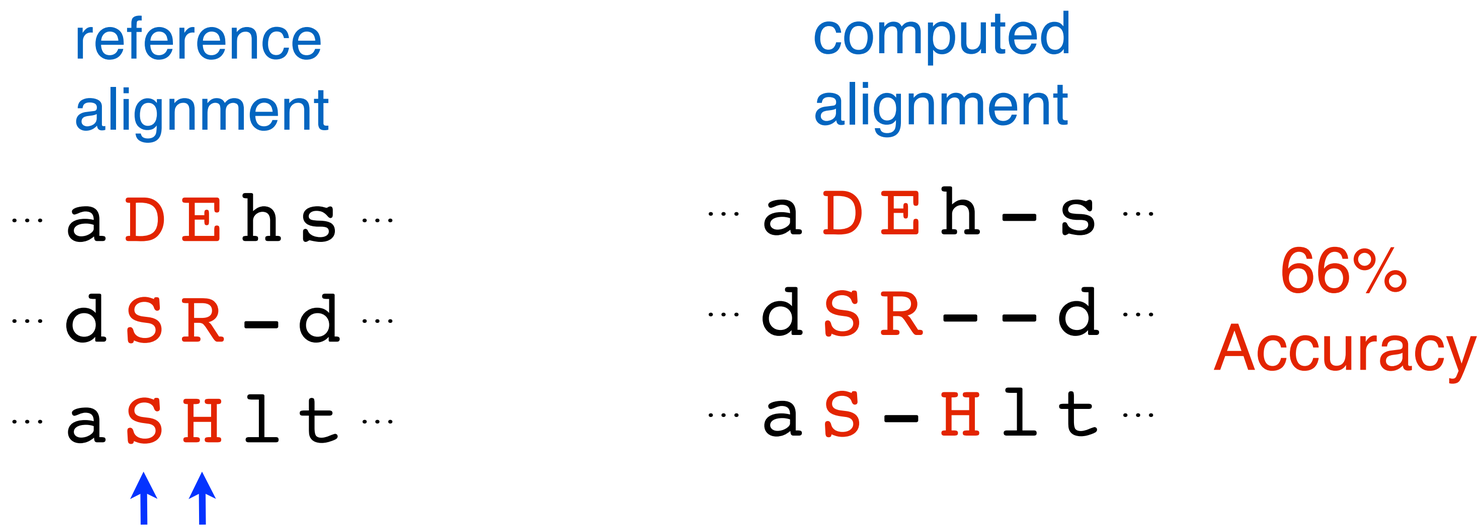
Alignment accuracy is measured with respect to a reference alignment.



- accuracy is the fraction of substitutions from the reference that are in the computed alignment,
- measured on the core columns of the reference.

Accuracy estimation

Alignment accuracy is measured with respect to a reference alignment.



- accuracy is the fraction of substitutions from the reference that are in the computed alignment,
- measured on the core columns of the reference.

Accuracy estimation

Our estimator `Facet` ("**F**eature-based **AC**curacy **E**s**T**imator")

- a polynomial on feature functions
- efficiently learns the coefficients from examples
- uses efficiently computed novel features

Feature functions are the key:
uninformative features → uninformative estimator

Accuracy estimation

The estimator $E(A)$ is a polynomial in the feature functions $f_i(A)$.

linear estimator

$$E(A) \quad := \quad \sum_i c_i \; f_i(A)$$

quadratic estimator

$$E(A) \quad := \quad \sum_i c_i \; f_i(A) + \sum_i \sum_j c_{ij} \; f_i(A) \; f_j(A)$$

Always linear in the coefficients.

Learning the estimator

We learn the estimator using examples consisting of

- an alignment, and
- its associated true accuracy.

Learning finds optimal coefficients that either fit

- accuracy values of the examples, or
- accuracy differences on pairs of examples,
- by solving a linear or quadratic program.

Learning the estimator

We learn the estimator using examples consisting of

- an alignment, and
- its associated true accuracy.

Learning finds optimal coefficients that either fit

- accuracy values of the examples, or
- accuracy differences on pairs of examples,
- by solving a linear or quadratic program.

Learning the estimator

Minimize

$$\sum_{a,b \in \text{examples}} w_{a,b}\, e_{a,b}$$

Subject to:

$$e_{a,b} \geq E(b) - E(a) = \sum_i c_i \left(f_i(b) - f_i(a)\right)$$

$$e_{a,b} \geq 0$$

$$\forall a, b \in \text{Examples} : Accuracy(a) > Accuracy(b)$$

Feature functions

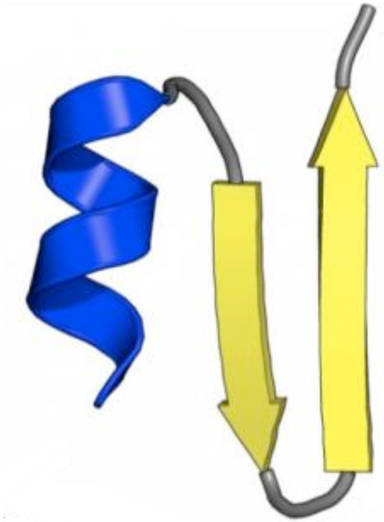
We use protein alignment feature functions that

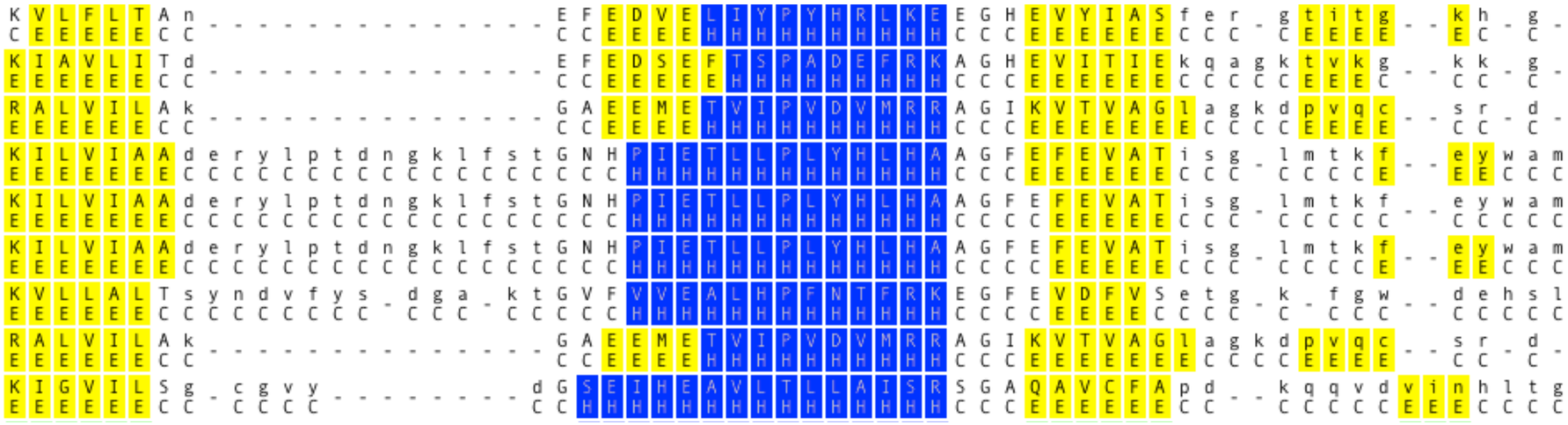
- are fast to evaluate,
- measure novel properties,
- use non-local information,
- involve secondary structure.

Feature functions

There are three types of secondary structure

- α-helix,
- β-strand,
- coil.





http://www.ebi.ac.uk/training/online/

Feature functions

Features based only on the input alignment

- Amino Acid Identity
- Average Substitution Score
- Information Content
- ...

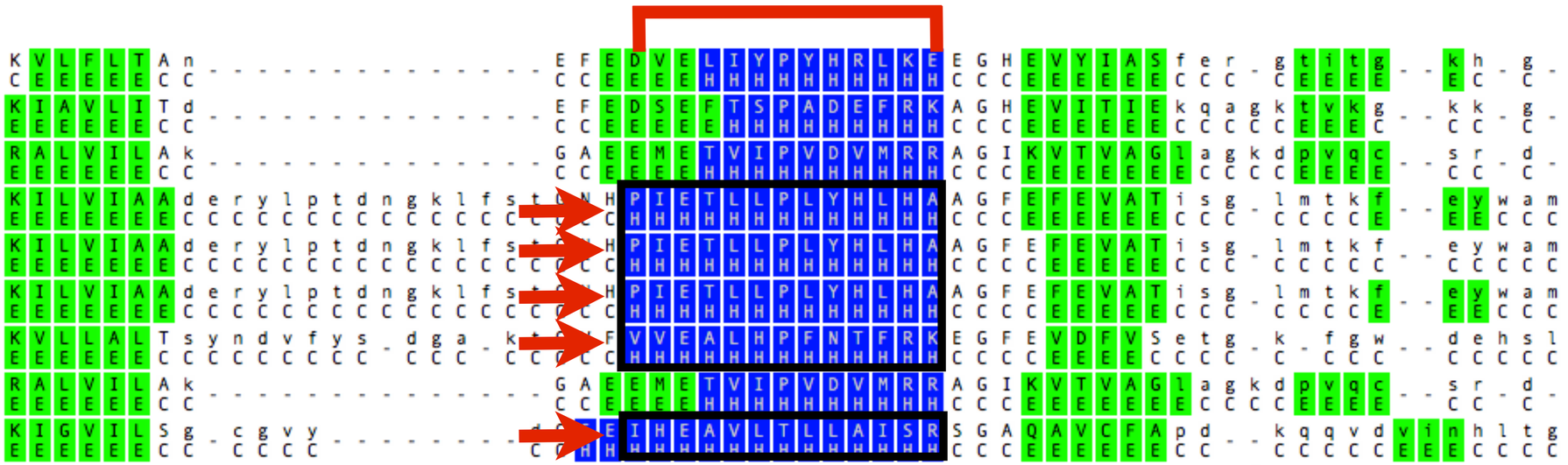
Features using predicted secondary structure

- Secondary Structure Percent Identity
- Secondary Structure Agreement
- Secondary Structure Blockiness
- ...

Secondary structure blockiness

A block B in alignment A is

- an interval of at least l columns,
- a subset of at least k rows,
- with the same secondary structure for all residues in B.

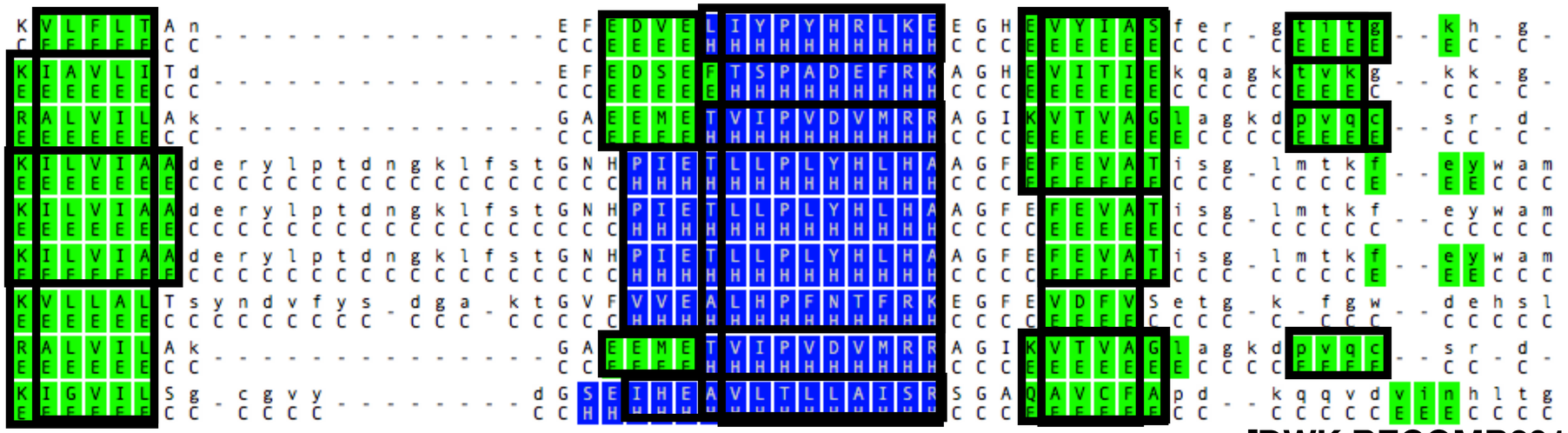


[DWK RECOMB2012/DK JCB2012]

Secondary structure blockiness

A block B in alignment A is

- an interval of at least l columns,
- a subset of at least k rows,
- with the same secondary structure for all residues in B.



[DWK RECOMB2012/DK JCB2012]

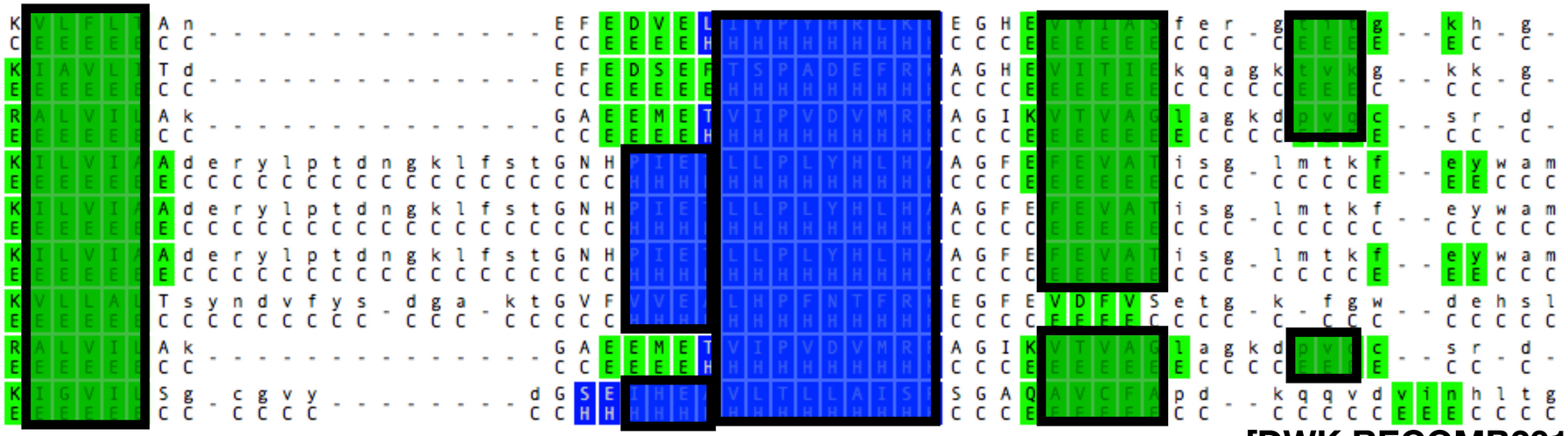
Secondary structure blockiness

A packing P for alignment A is

- a set of blocks from A,
- whose columns are disjoint.

The value of P is the number of substitutions it contains.

The Blockiness feature is the maximum value of any packing.



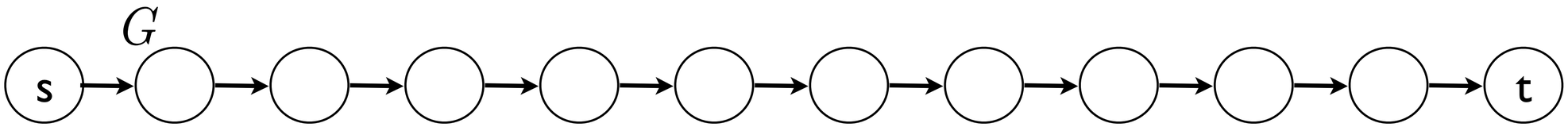
[DWK RECOMB2012/DK JCB2012]

Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

Algorithm



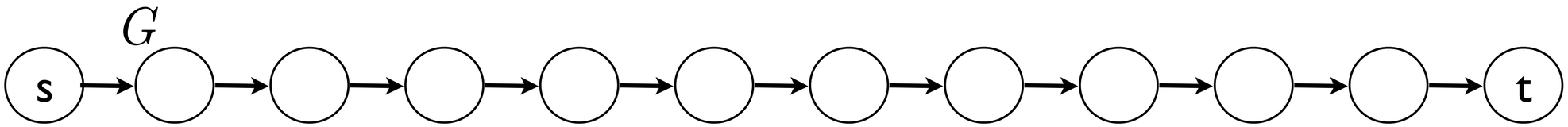
-	E	F	E	D	S	E	F	T	S
-	G	A	E	E	M	E	T	V	I
t	G	N	H	P	I	E	T	L	L
t	G	N	H	P	I	H	T	L	L

Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

Algorithm



minimum width $l = 3$

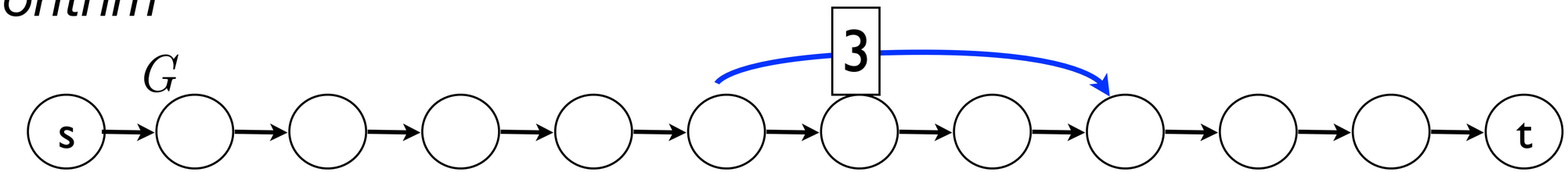
-	E	F	E	D	S	E	F	T	S
-	G	A	E	E	M	E	T	V	I
t	G	N	H	P	I	E	T	L	L
t	G	N	H	P	I	H	T	L	E

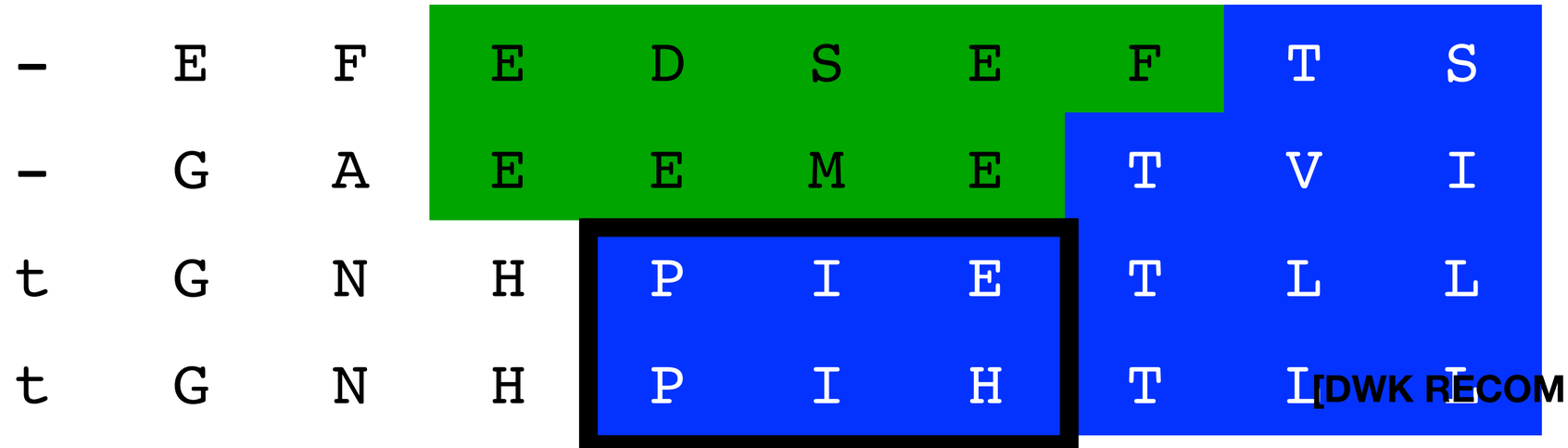
Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

Algorithm



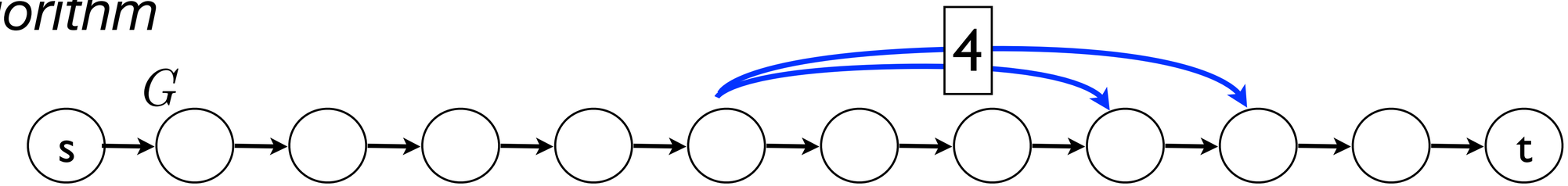


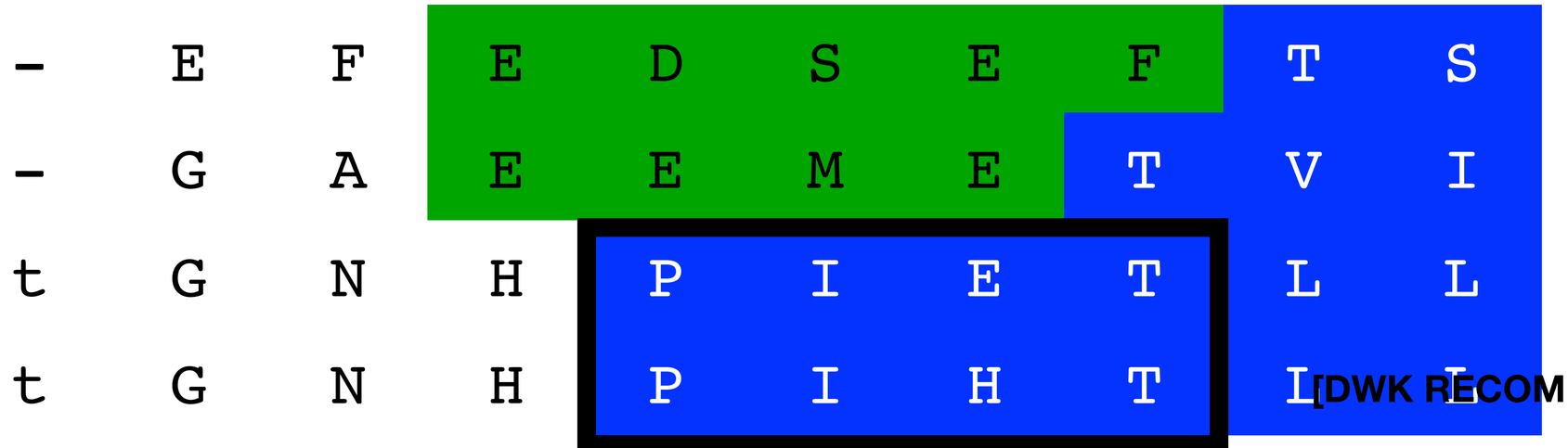
Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

Algorithm



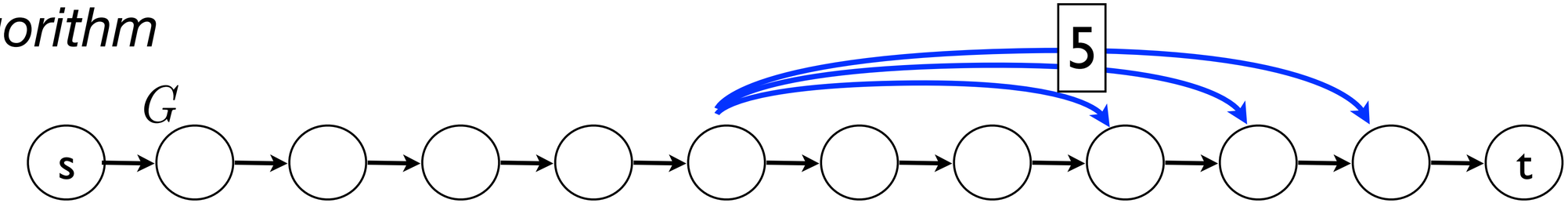


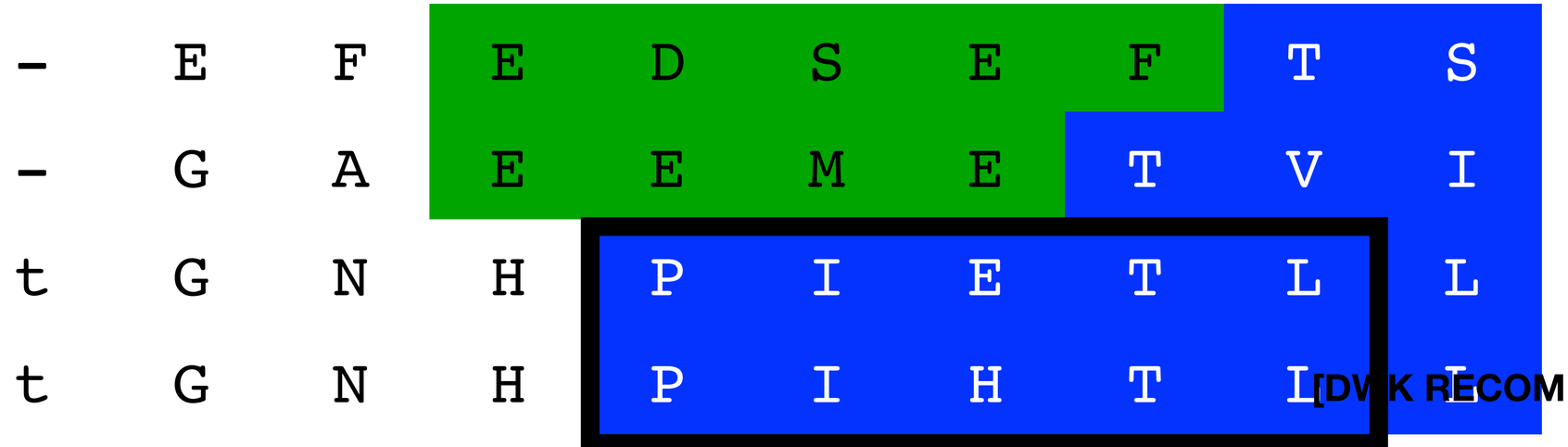
Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

Algorithm



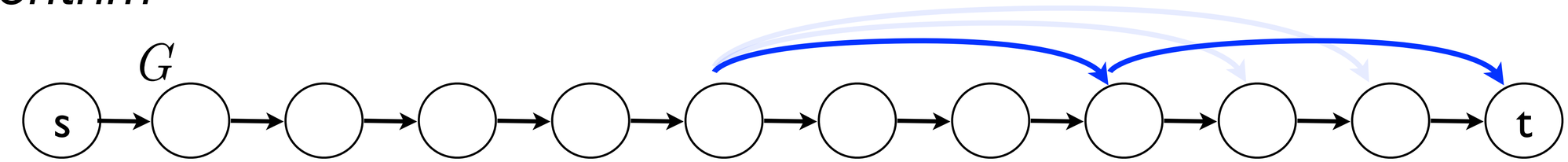


Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

Algorithm



minimum width $l = 3$

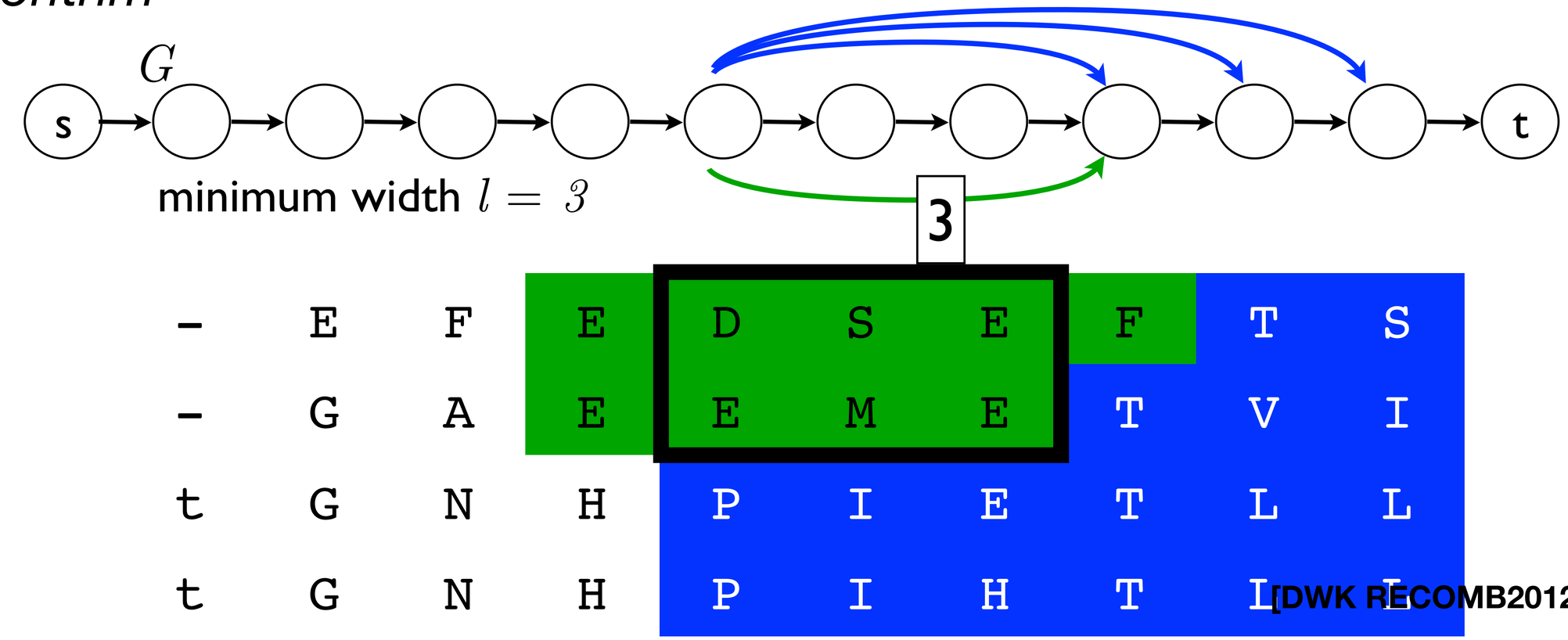
-	E	F	E	D	S	E	F	T	S
-	G	A	E	E	M	E	T	V	I
t	G	N	H	P	I	E	T	L	L
t	G	N	H	P	I	H	T	I	E

Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

Algorithm

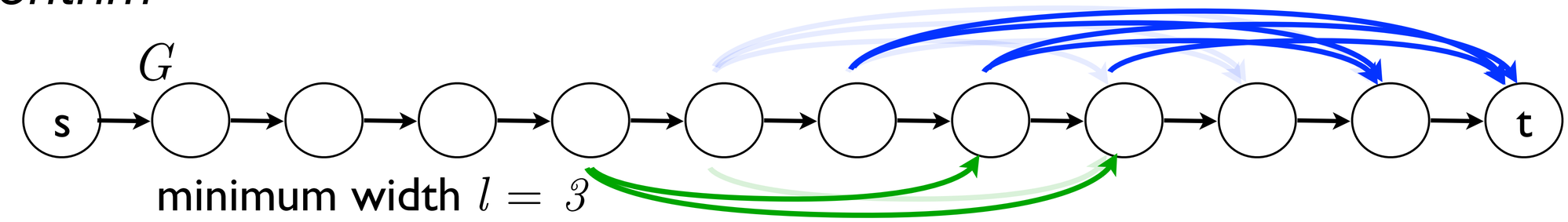


Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

Algorithm



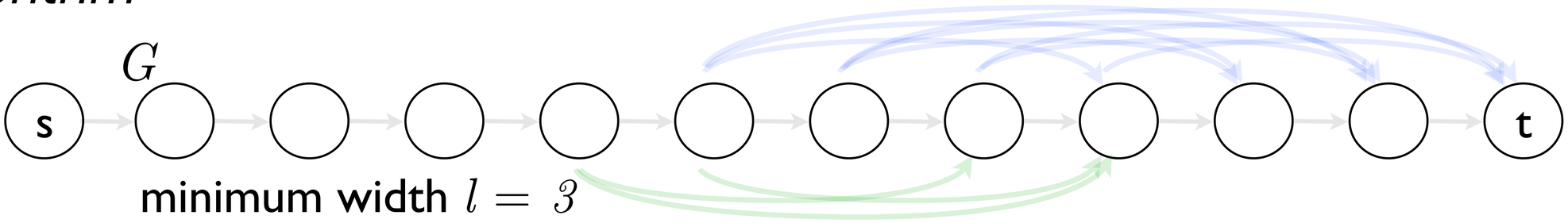

```
-   E   F   E   D   S   E   F   T   S
-   G   A   E   E   M   E   T   V   I
t   G   N   H   P   I   E   T   L   L
t   G   N   H   P   I   H   T   L   E
```

Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

Algorithm



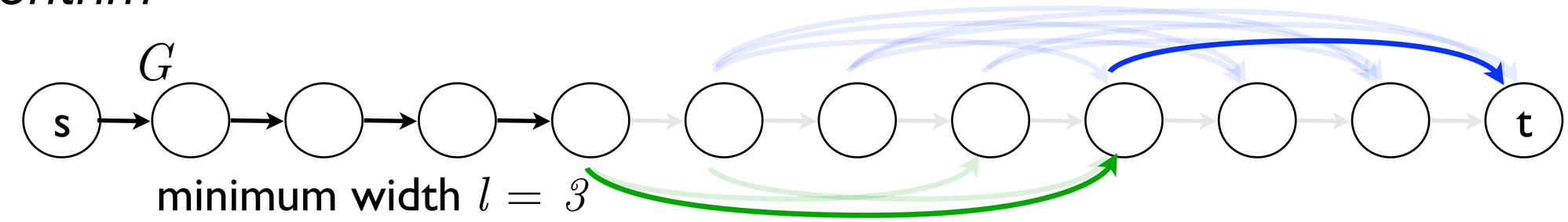
-	E	F	E	D	S	E	F	T	S
-	G	A	E	E	M	E	T	V	I
t	G	N	H	P	I	E	T	L	L
t	G	N	H	P	I	H	T	L	I

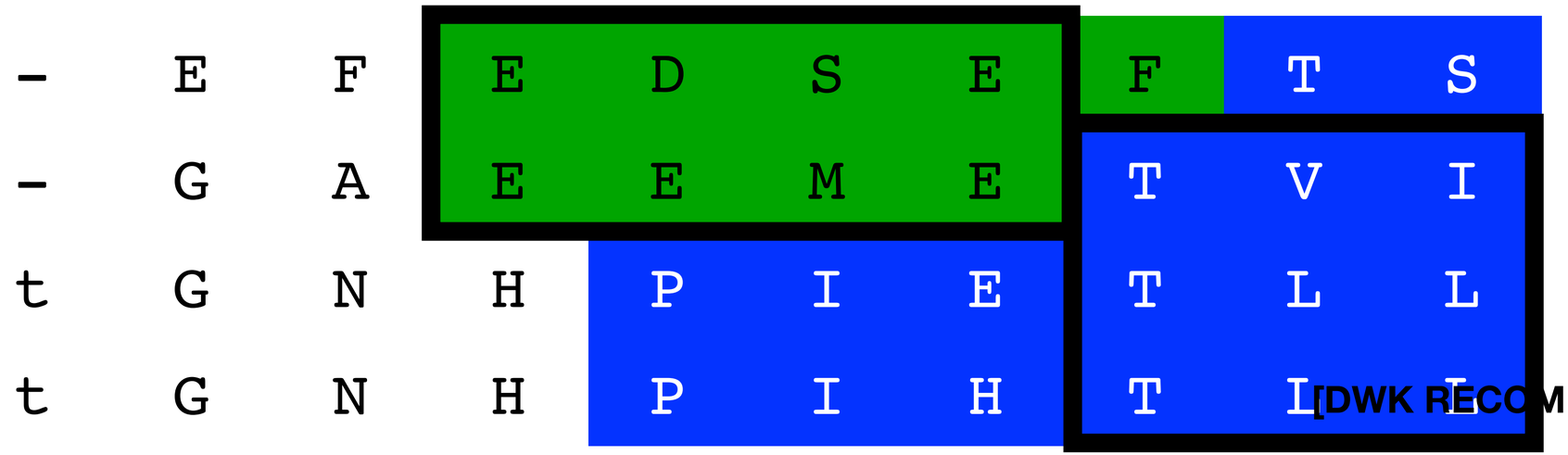
Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

Algorithm



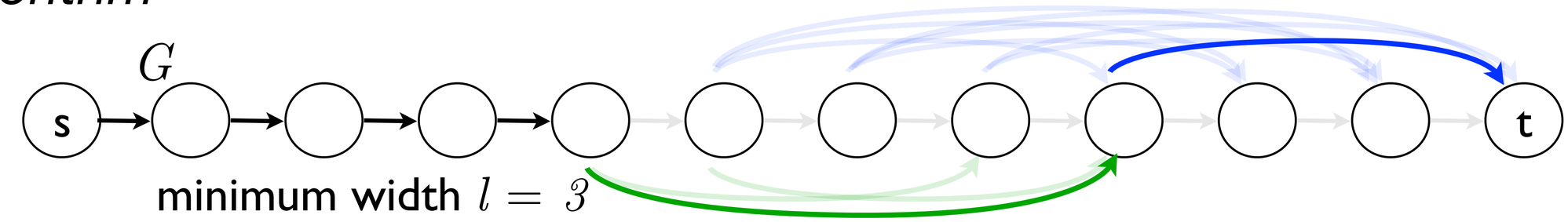


Secondary Structure Blockiness

Theorem (Evaluating Blockiness)

Blockiness can be computed in $O(mn)$ time,
for an alignment with m rows and n columns.

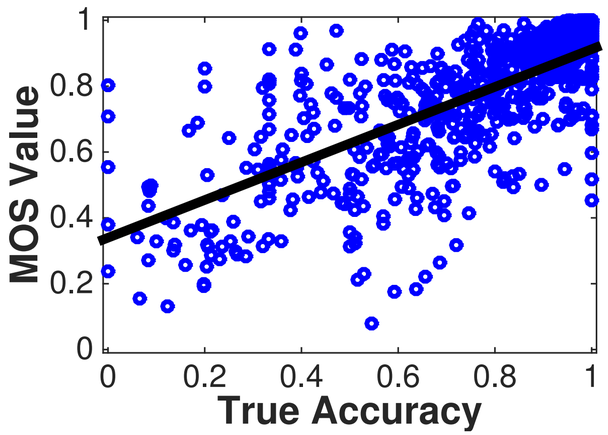
Algorithm



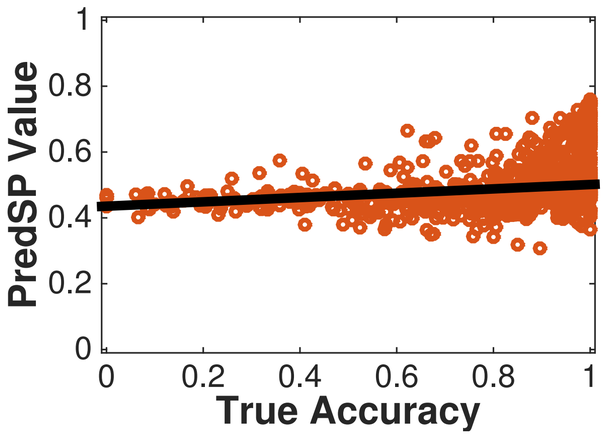
- Graph construction takes $O(mn)$ time.
- Graph has $O(n)$ nodes, $O(ln)$ edges
- Longest path takes $O(n)$ time.

Accuracy estimation

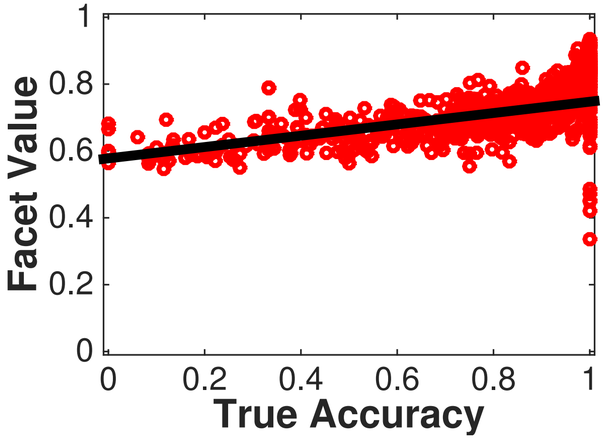
For parameter advising, an estimator should have high slope and low spread.



high slope,
high spread



low slope,
low spread



medium slope,
low spread

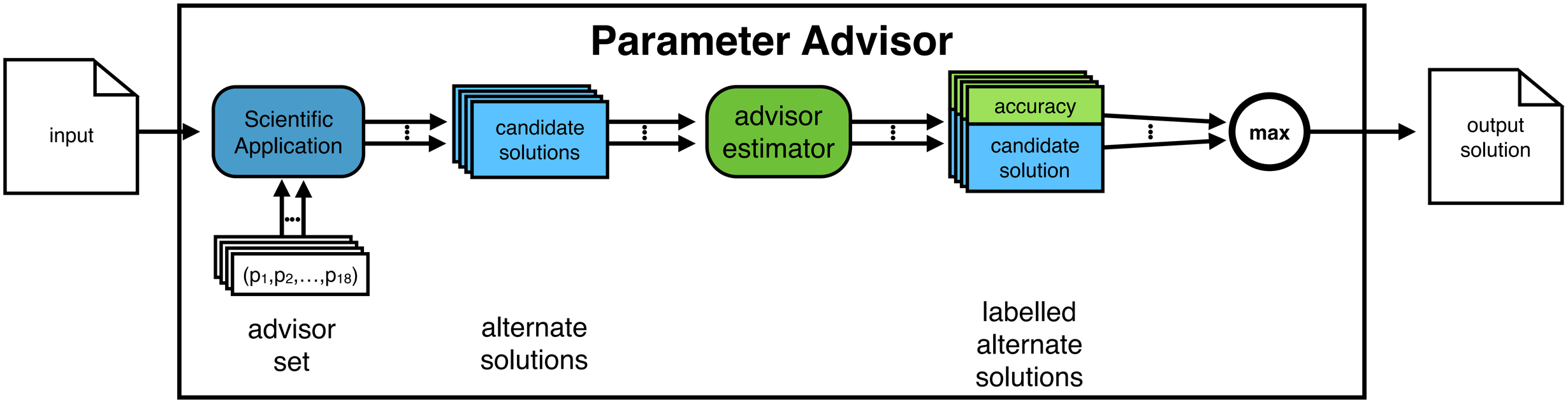
Facet's slope and spread is best for advising

[MOS: Lassmann and Sonnhammer, NAR 2005] [PredSP: Ahola, et al. Bioinformatics 2008] [DWK RECOMB2012/DK JCB2012]

Parameter advising

Components of an advisor:

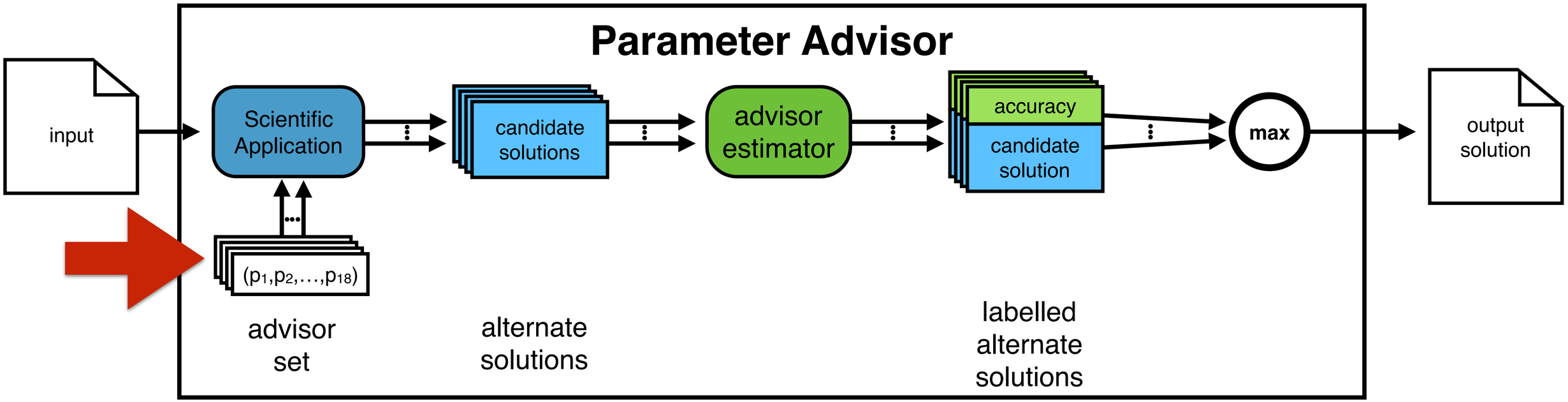
- An advisor set of parameter choice vectors.
- An advisor estimator to rank solutions.



Parameter advising

Components of an advisor:

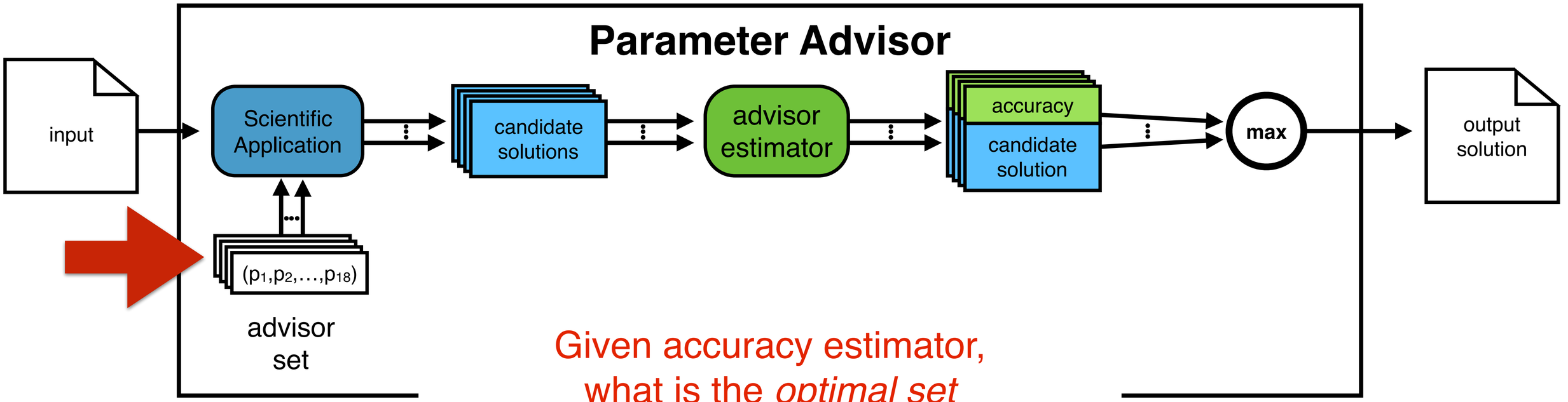
- An advisor set of parameter choice vectors.
- An advisor estimator to rank solutions.



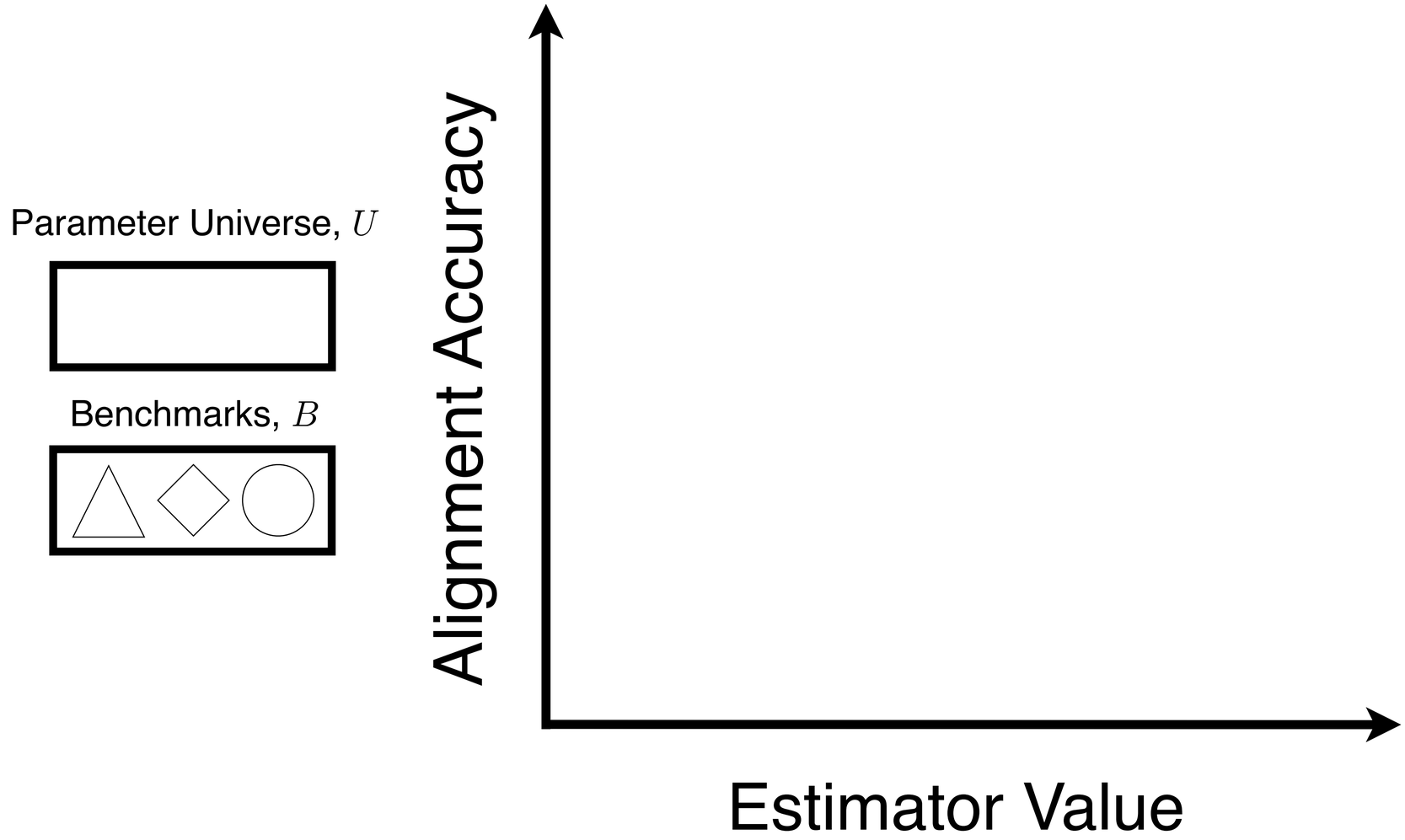
Parameter advising

Components of an advisor:

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

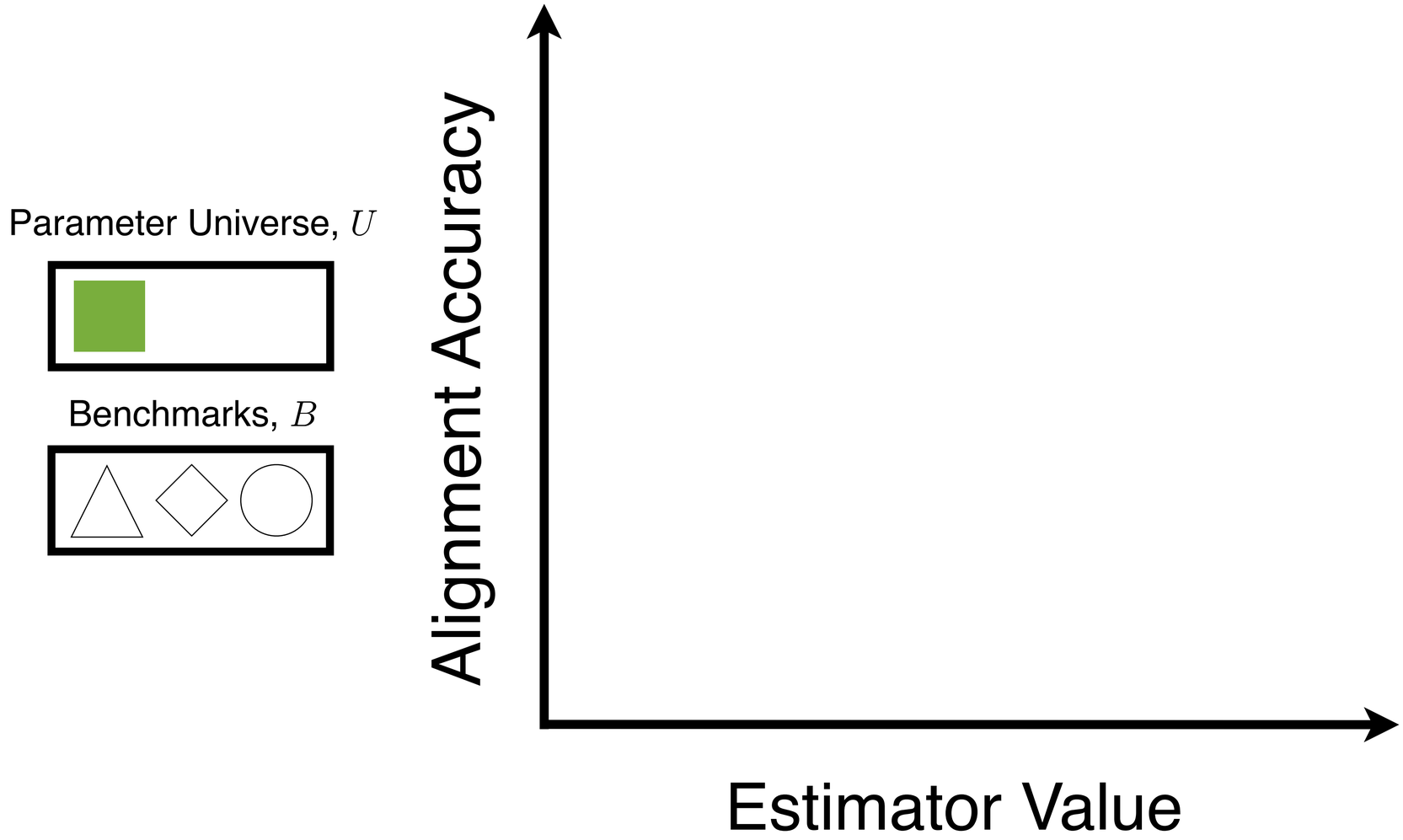


Advisor Set problem



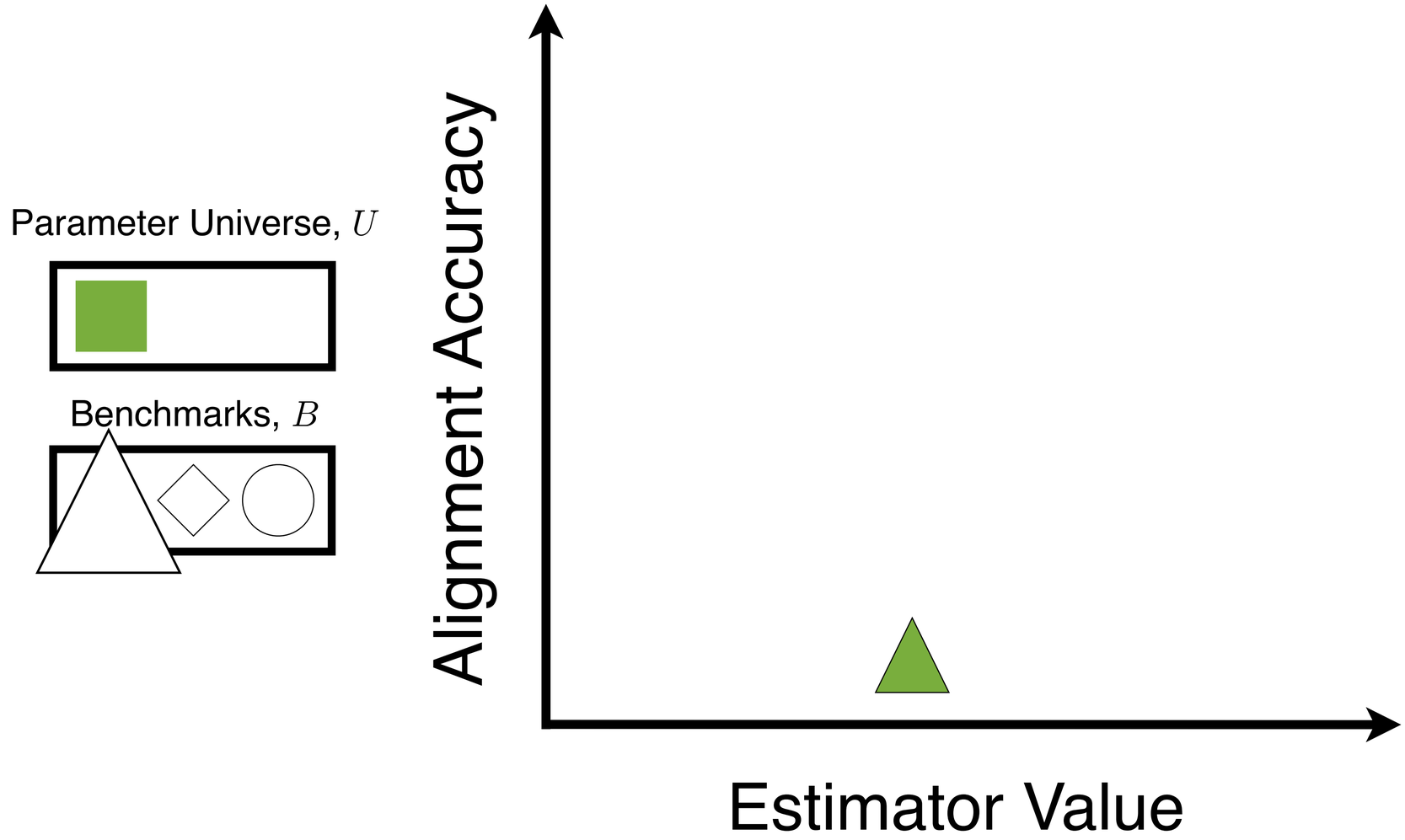
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



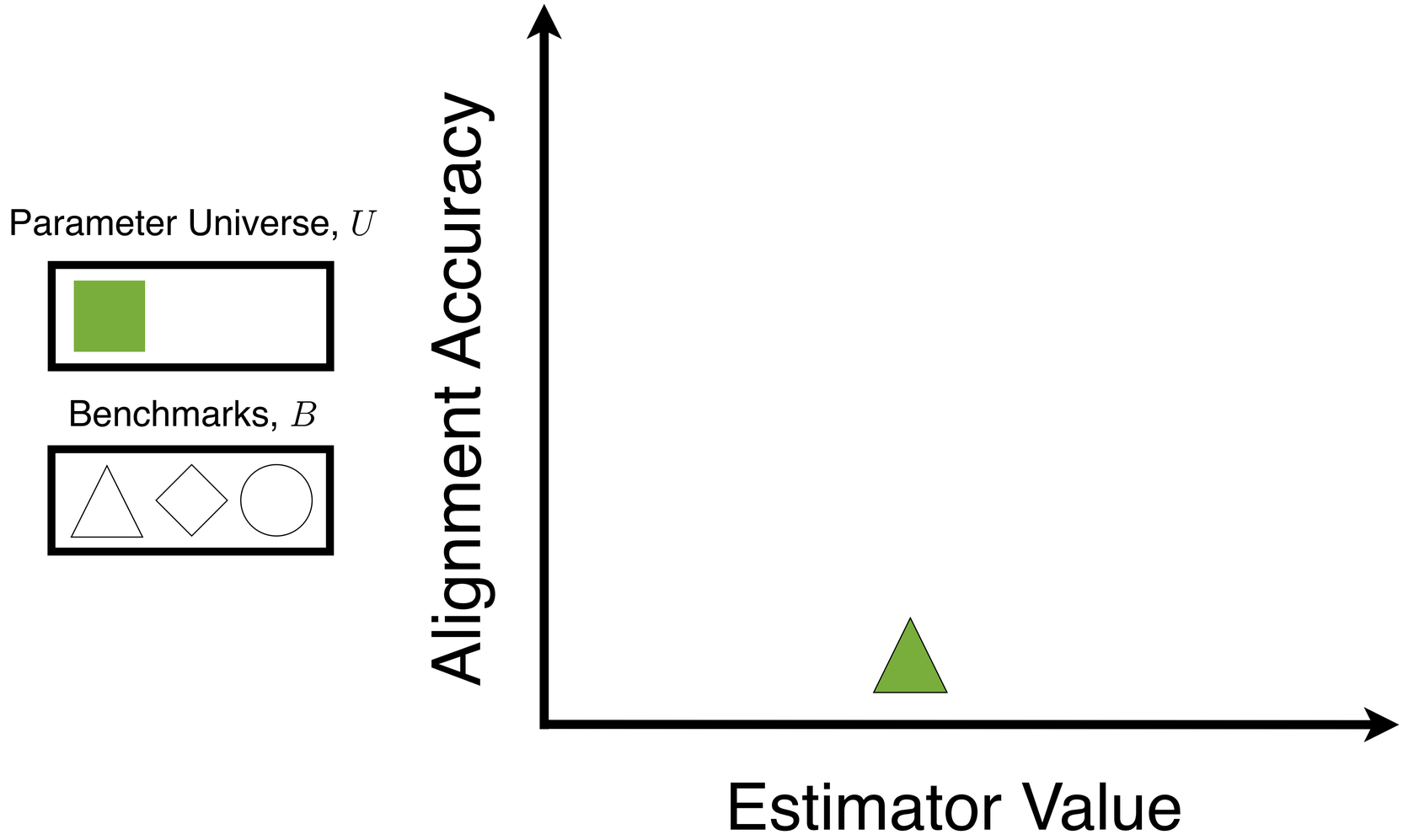
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



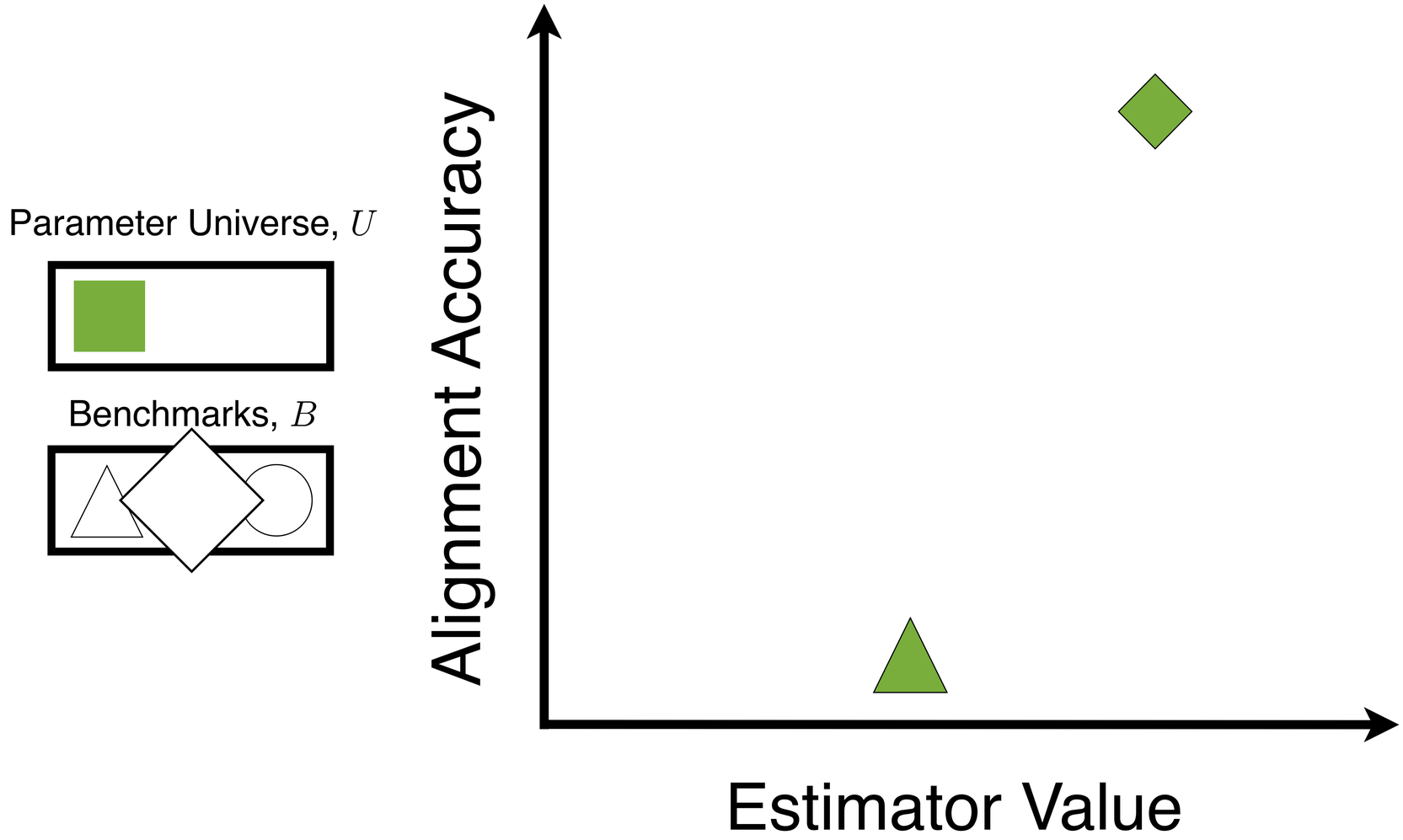
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



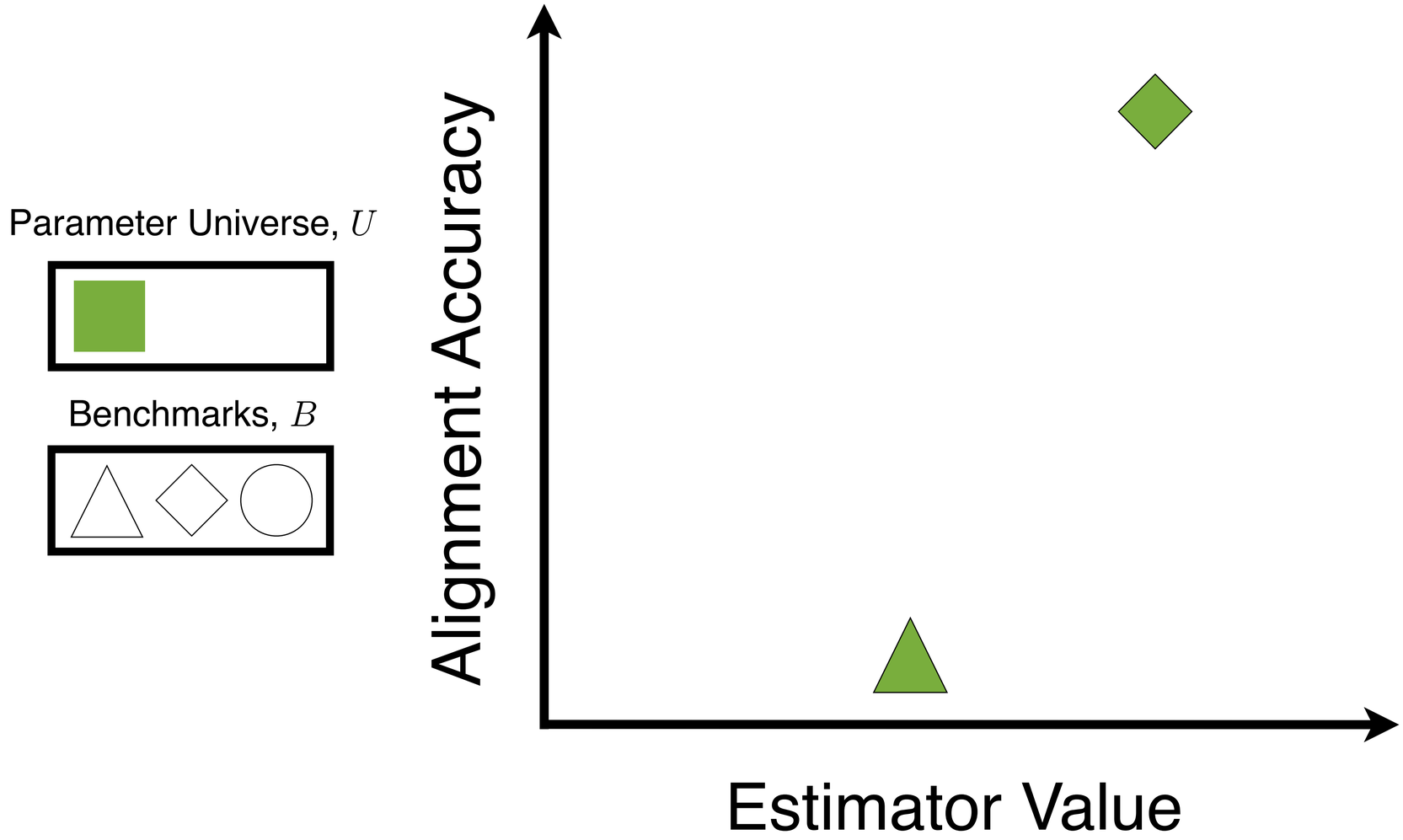
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



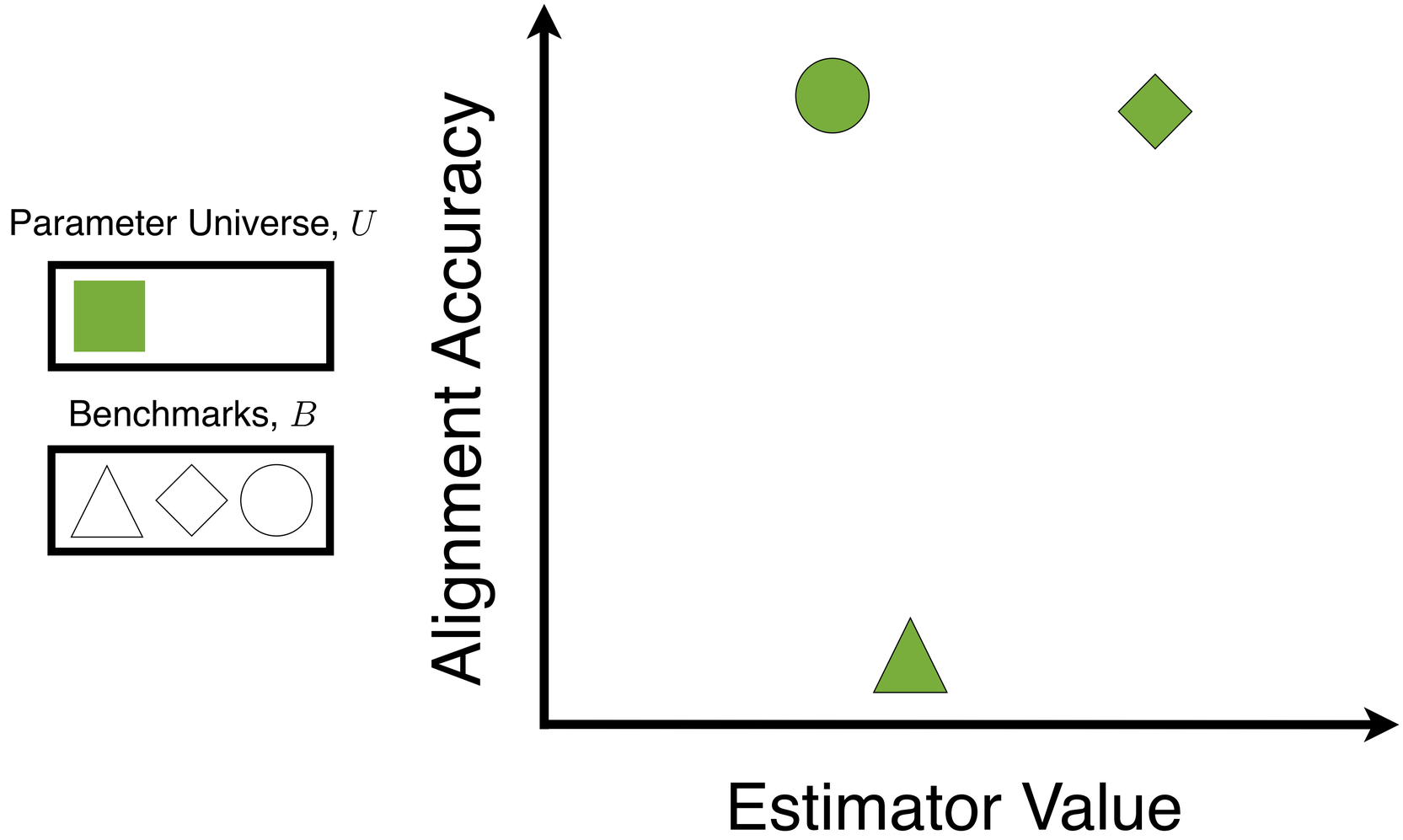
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



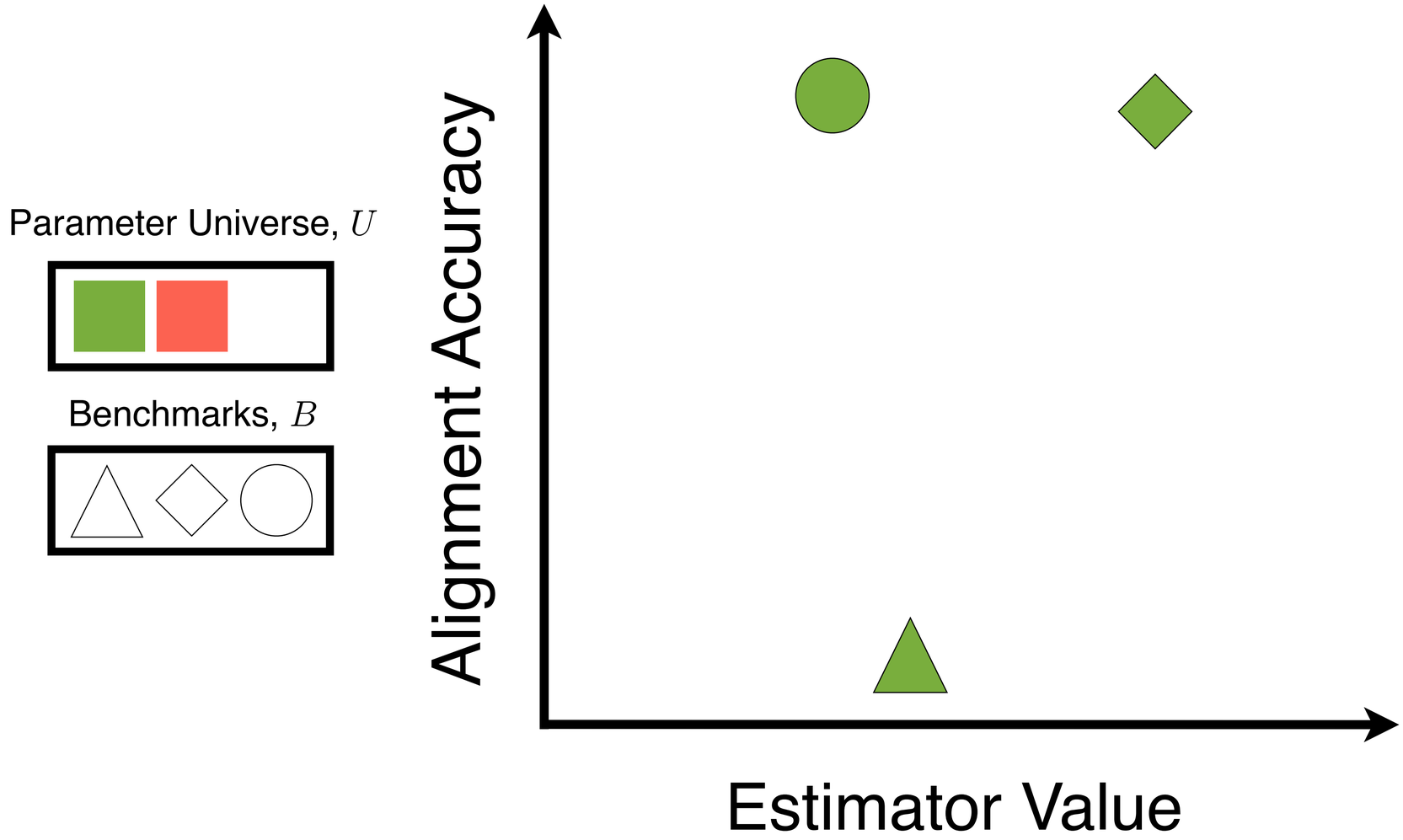
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



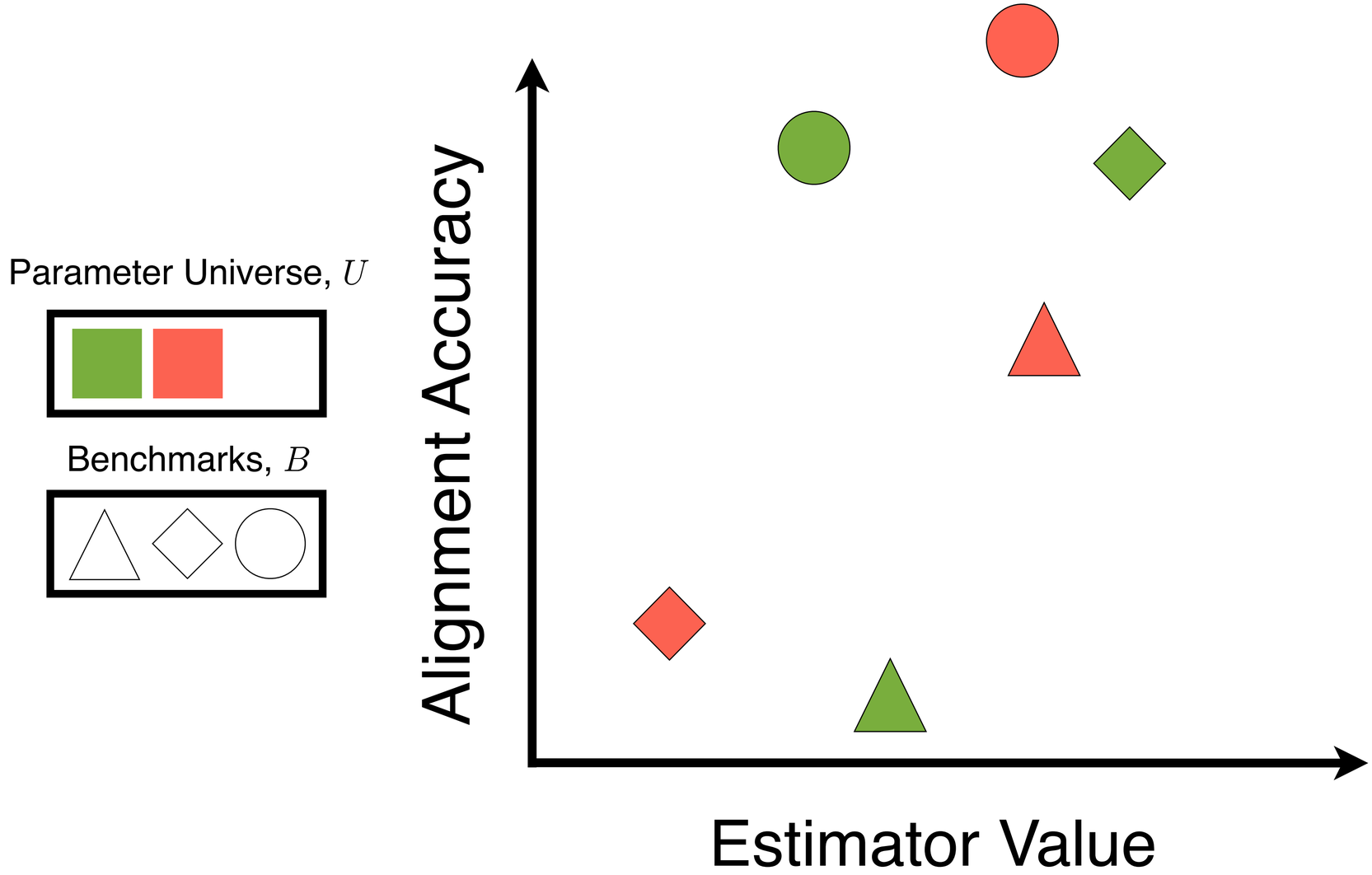
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



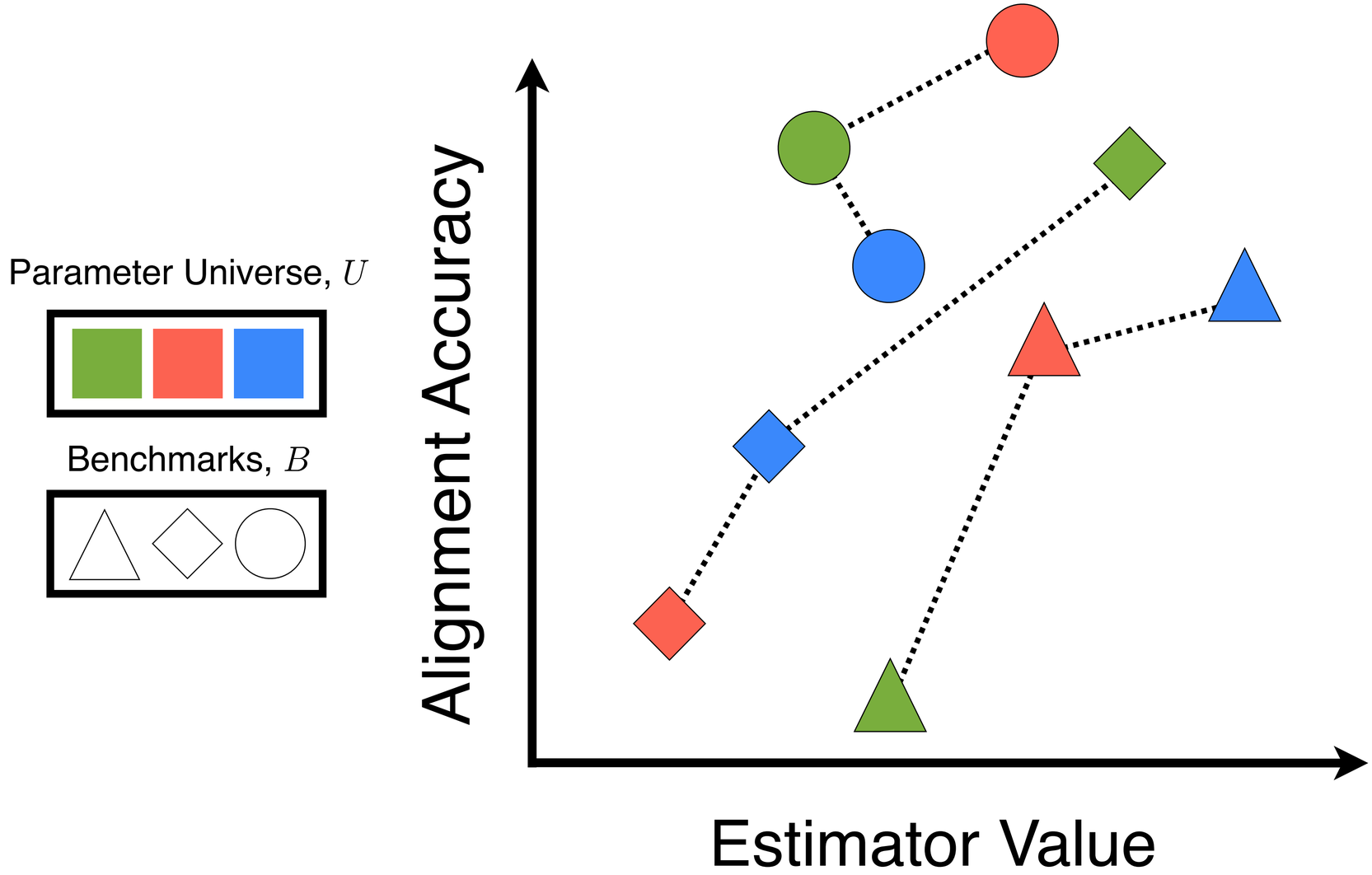
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



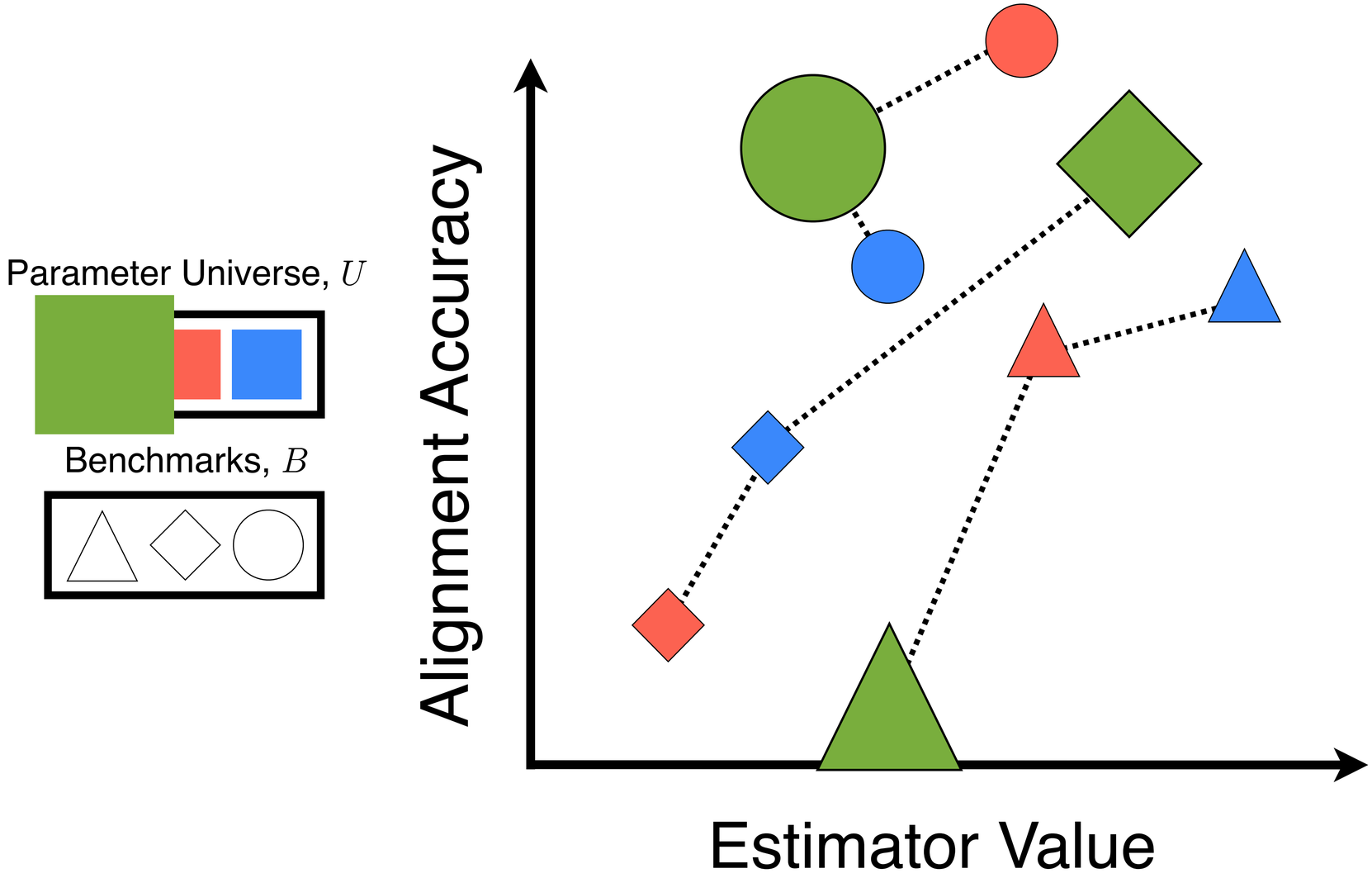
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



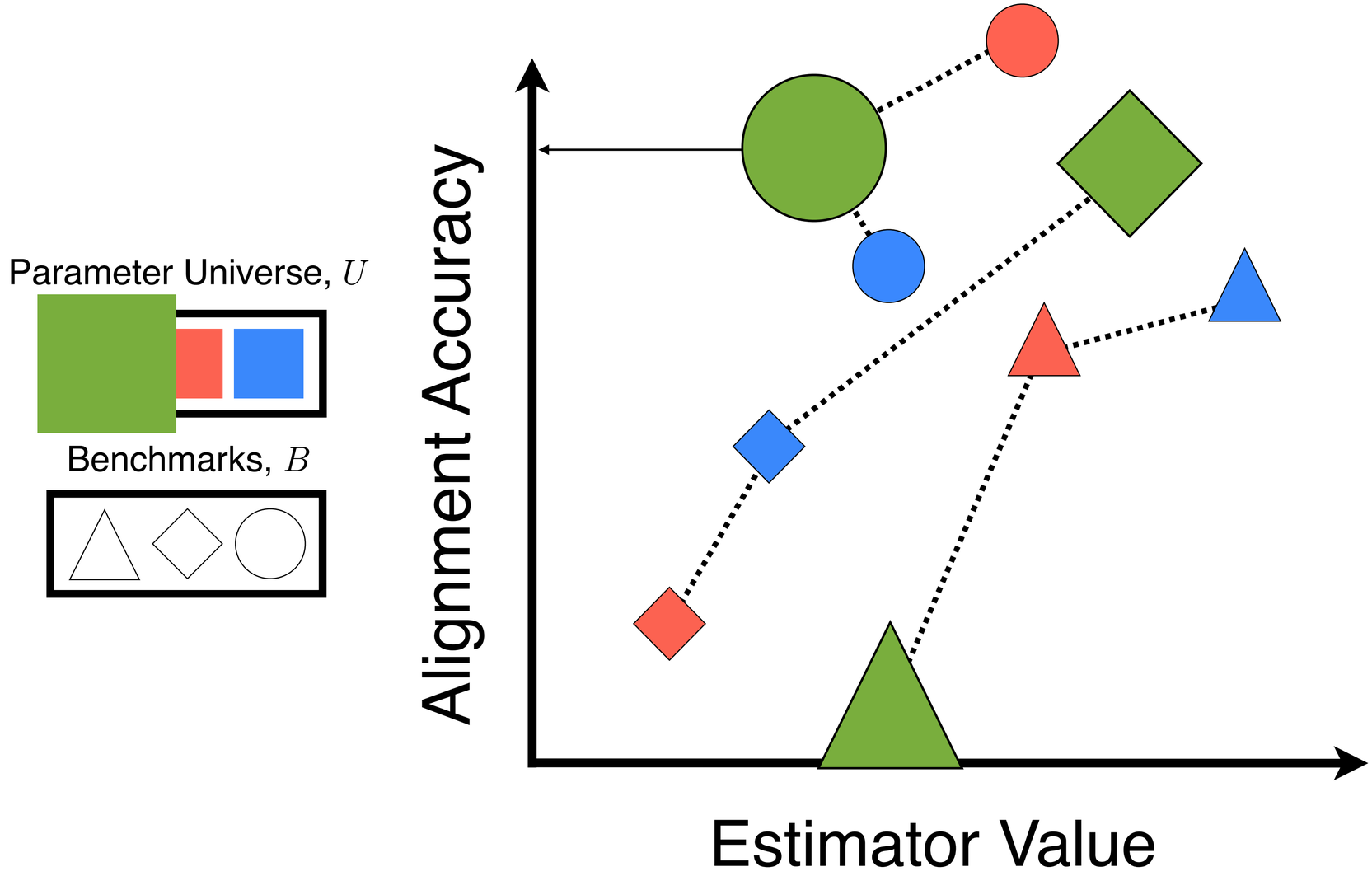
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



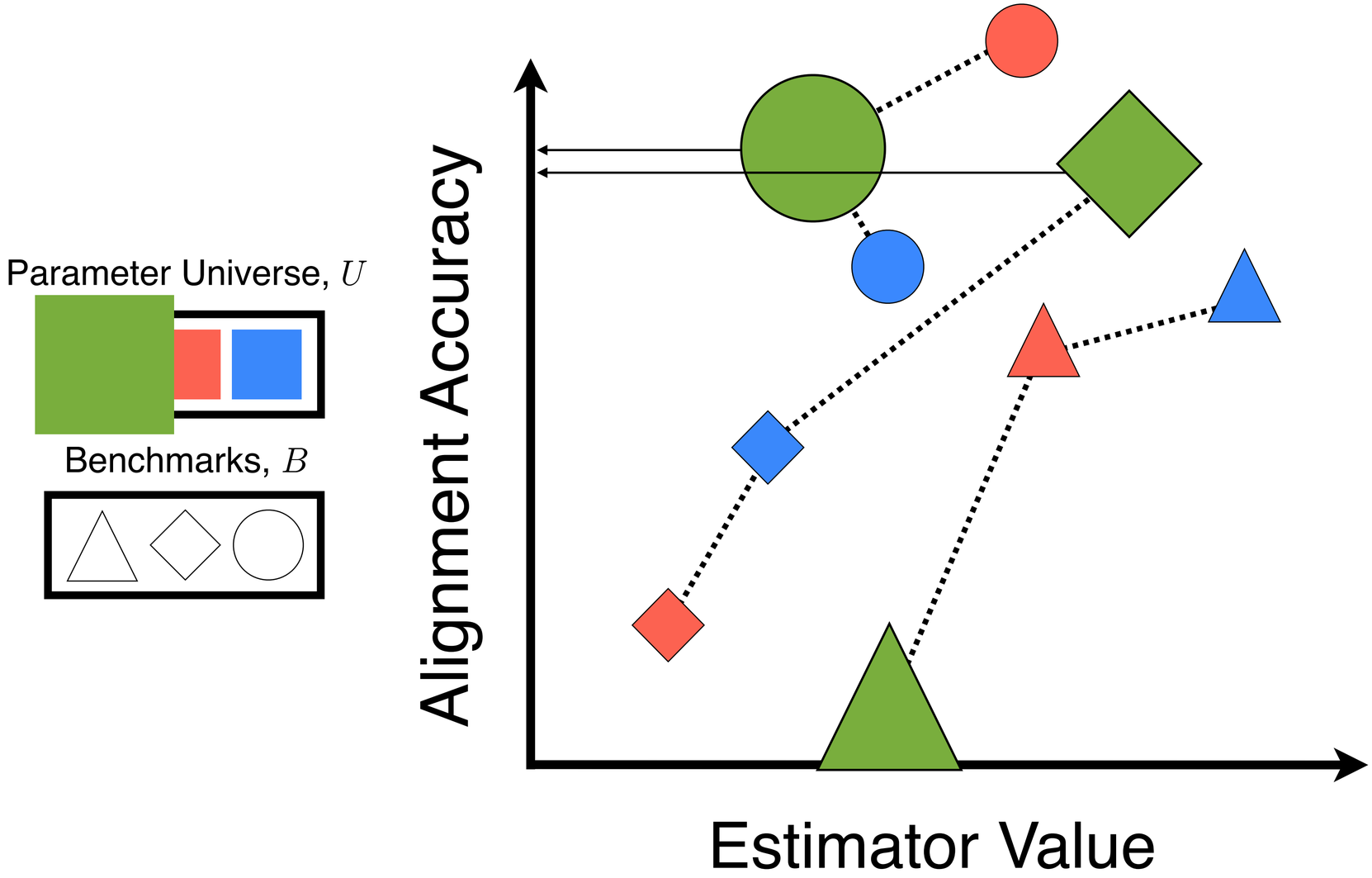
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem

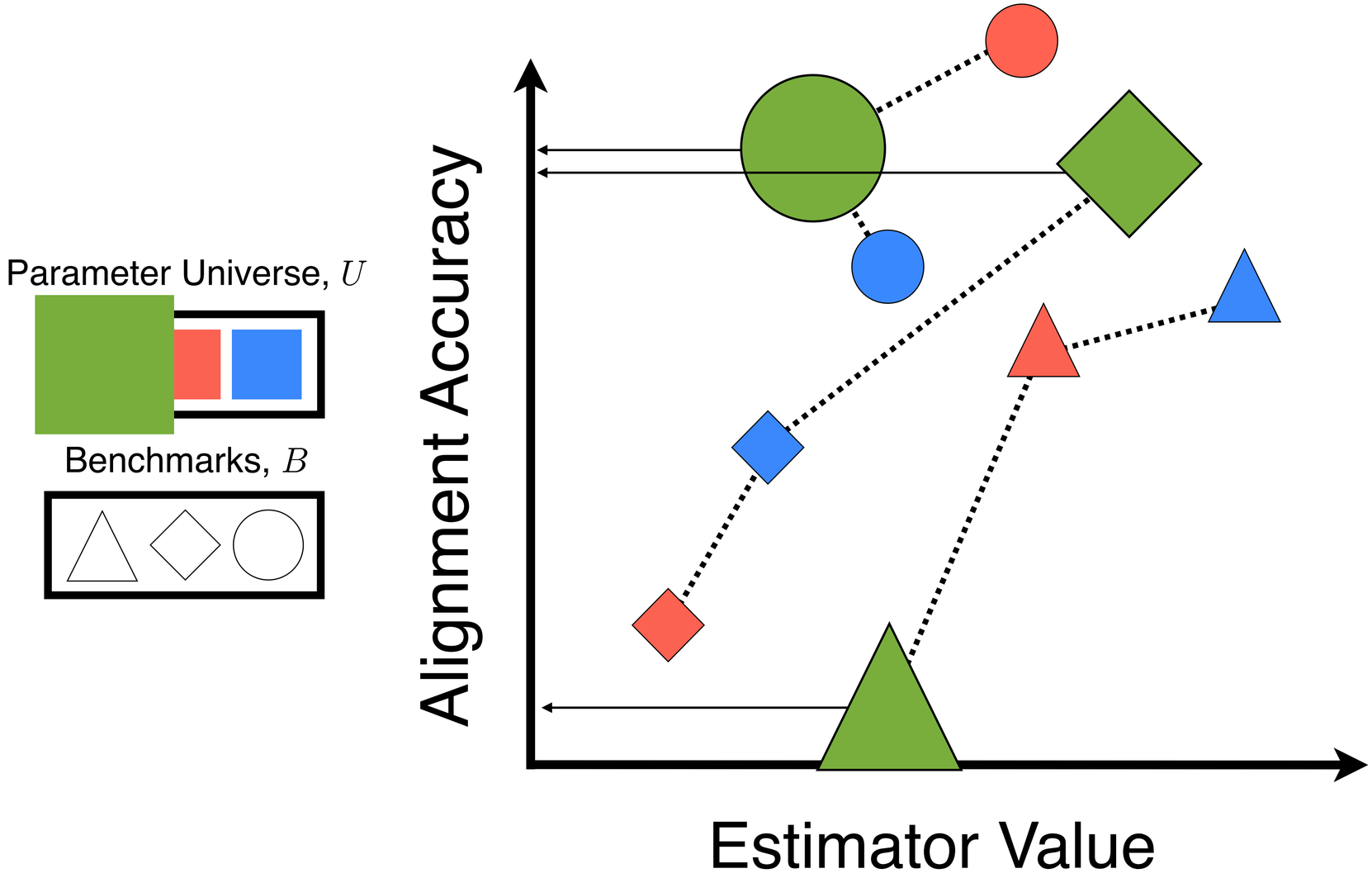


[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem

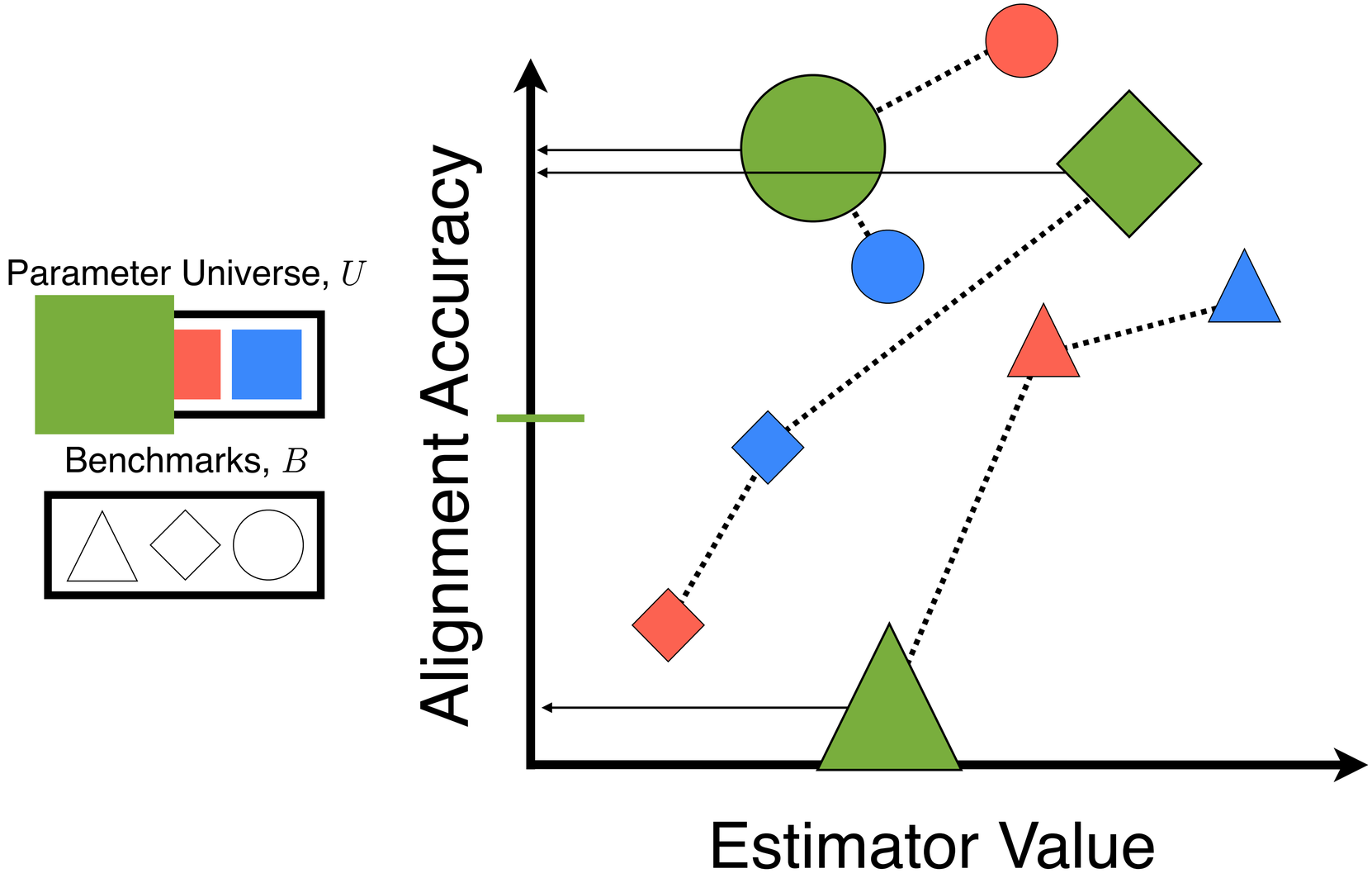


Advisor Set problem

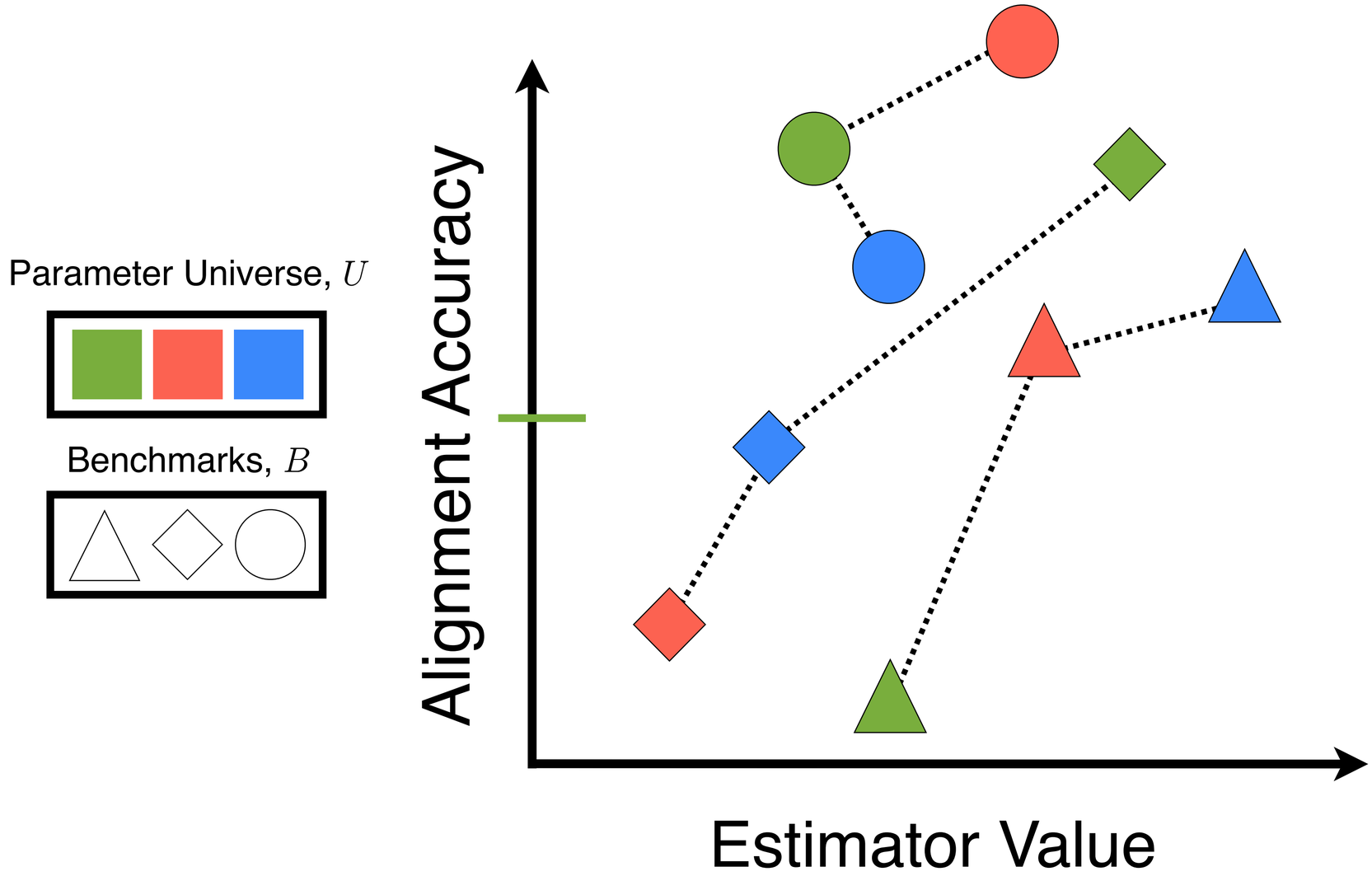


[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem

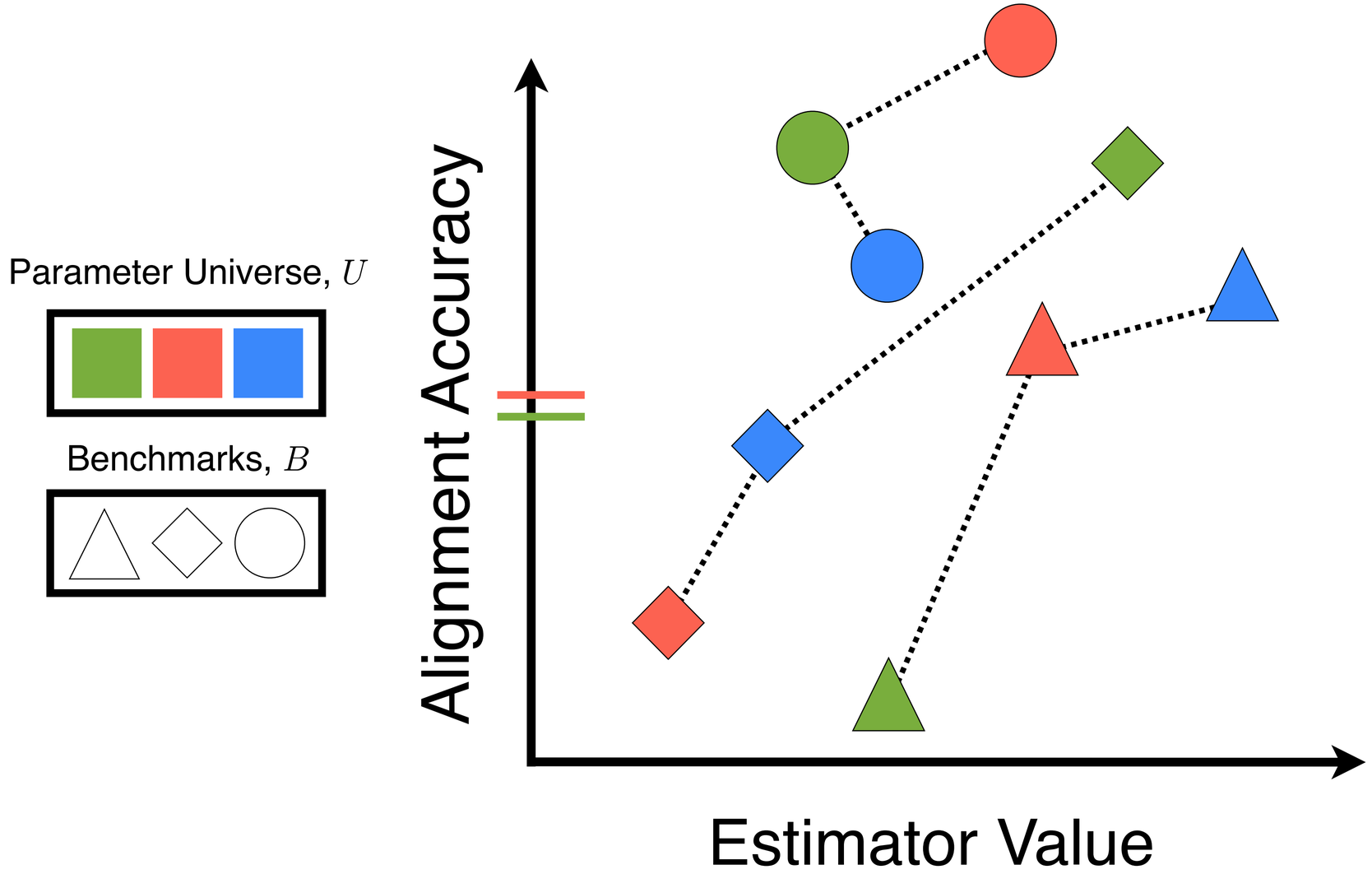


Advisor Set problem

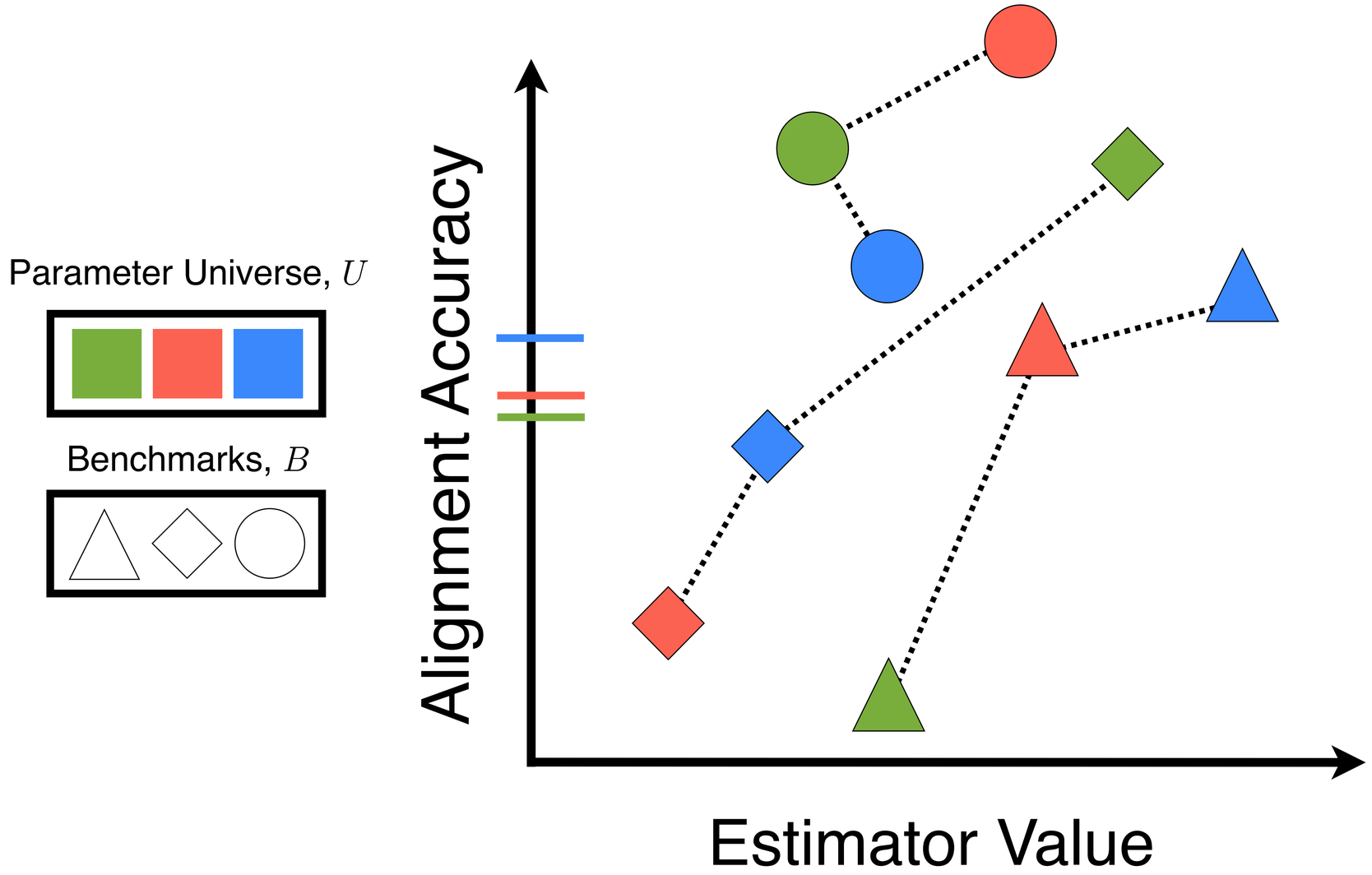


[DK ACM-BCB 2014, DK TCBB 2016]

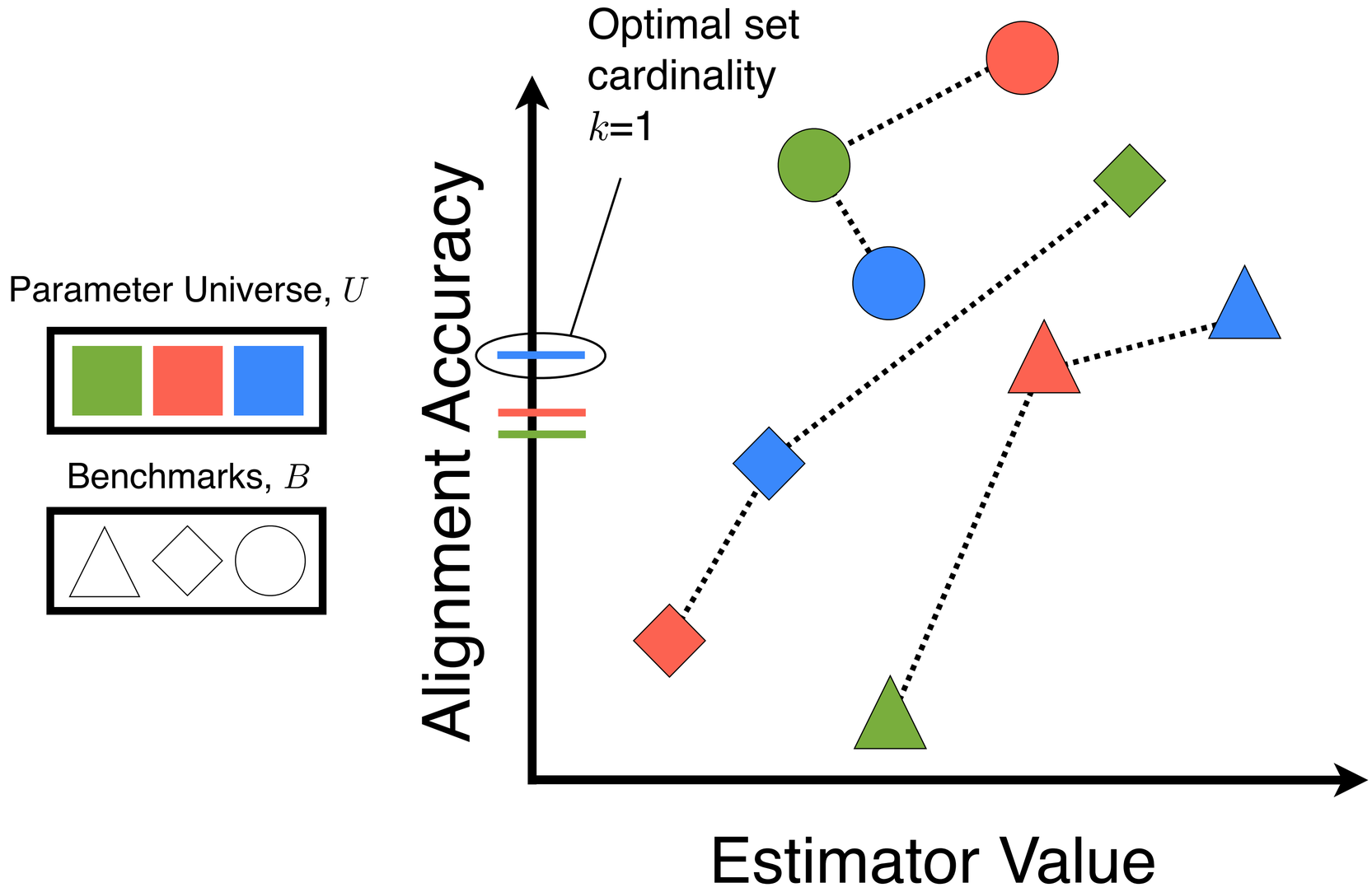
Advisor Set problem



Advisor Set problem

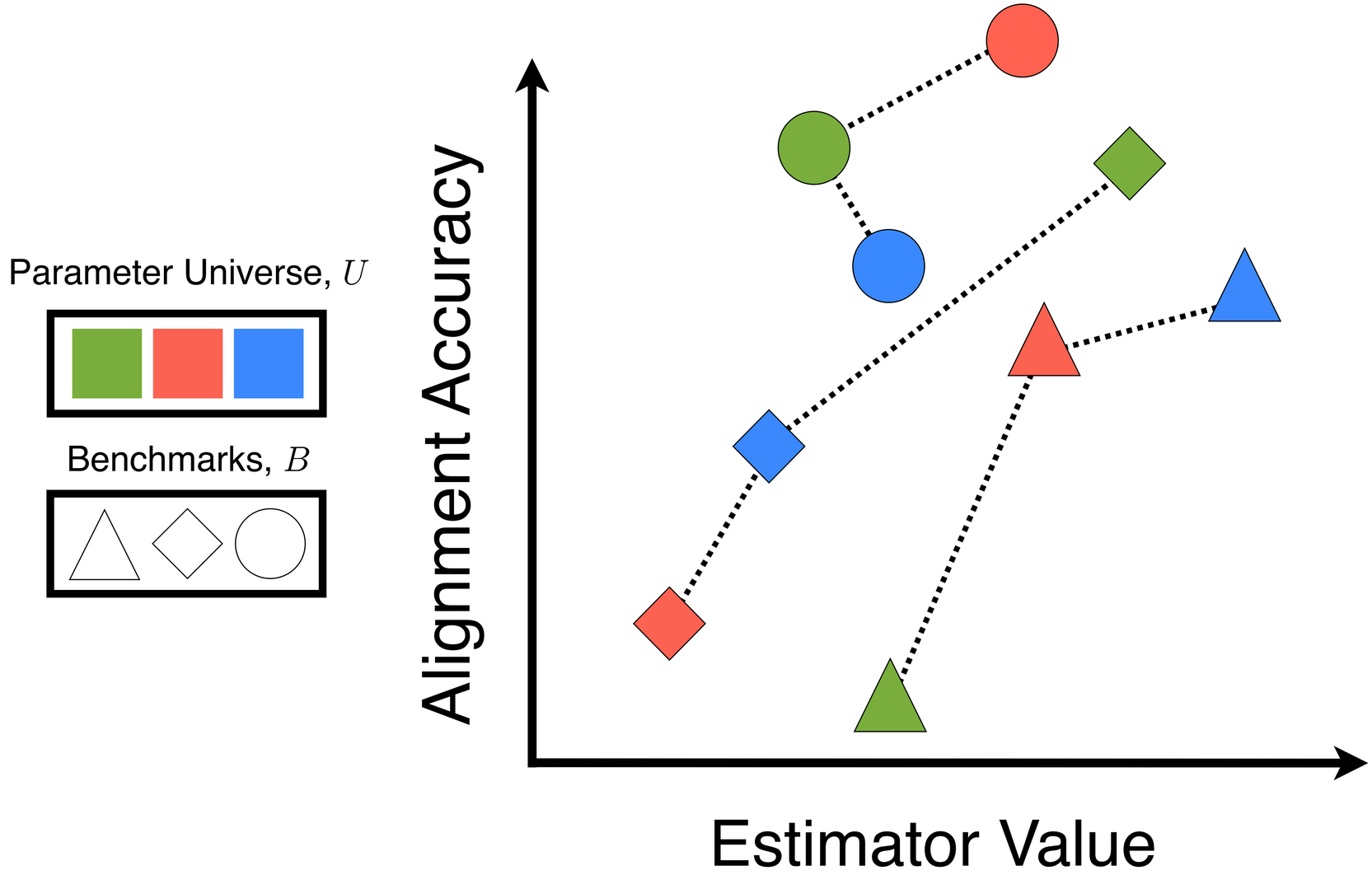


Advisor Set problem



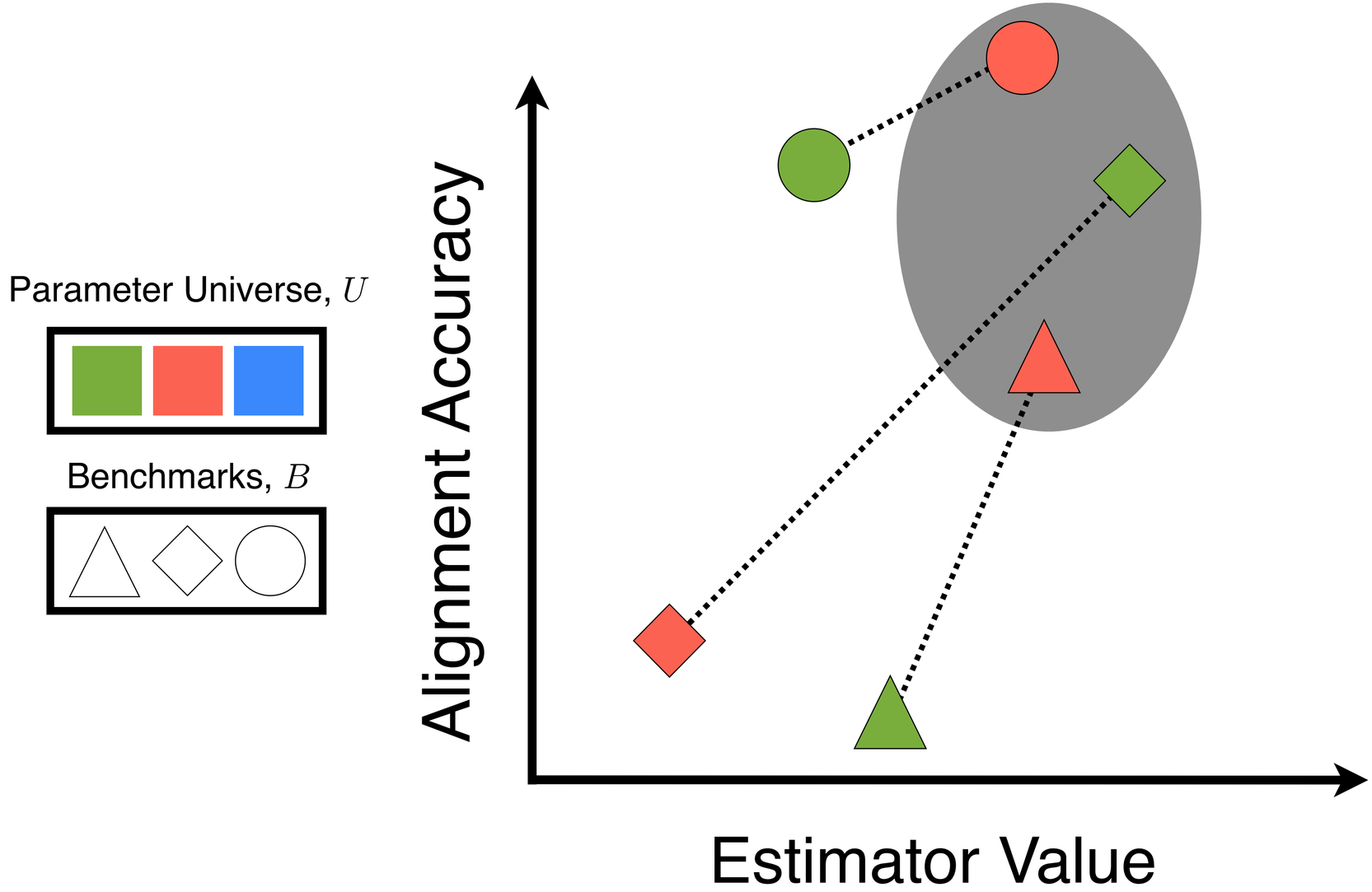
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



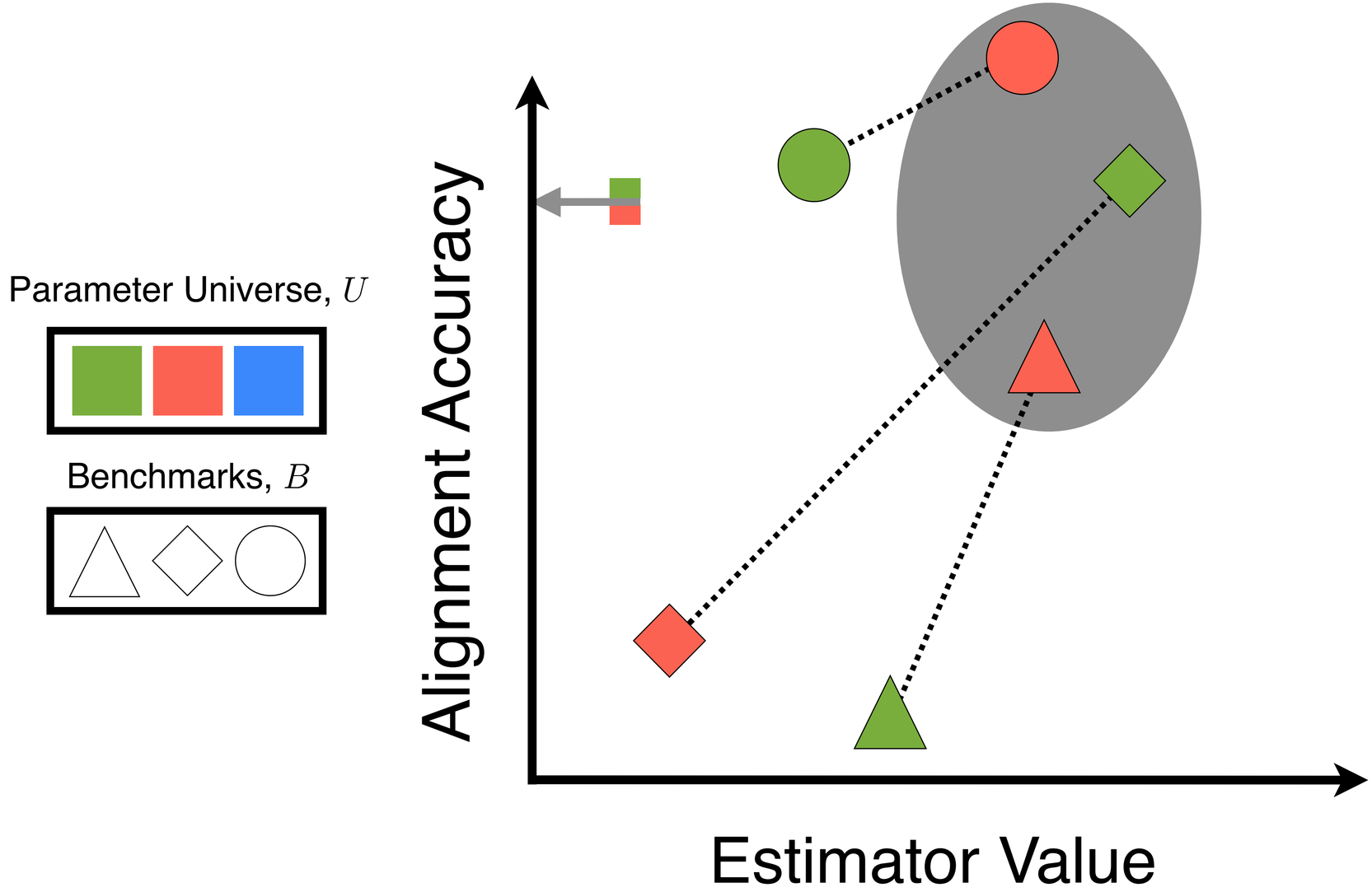
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



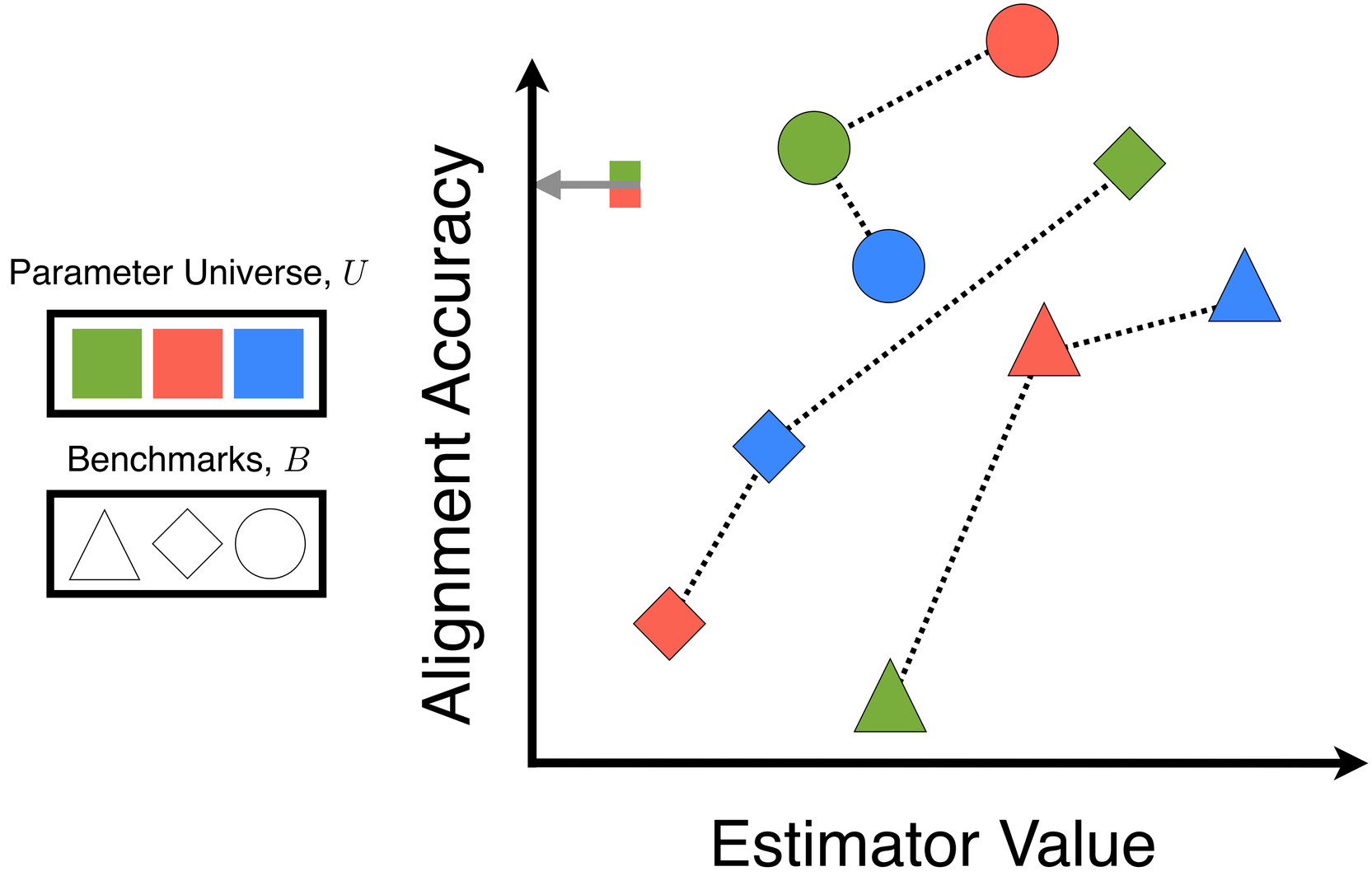
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



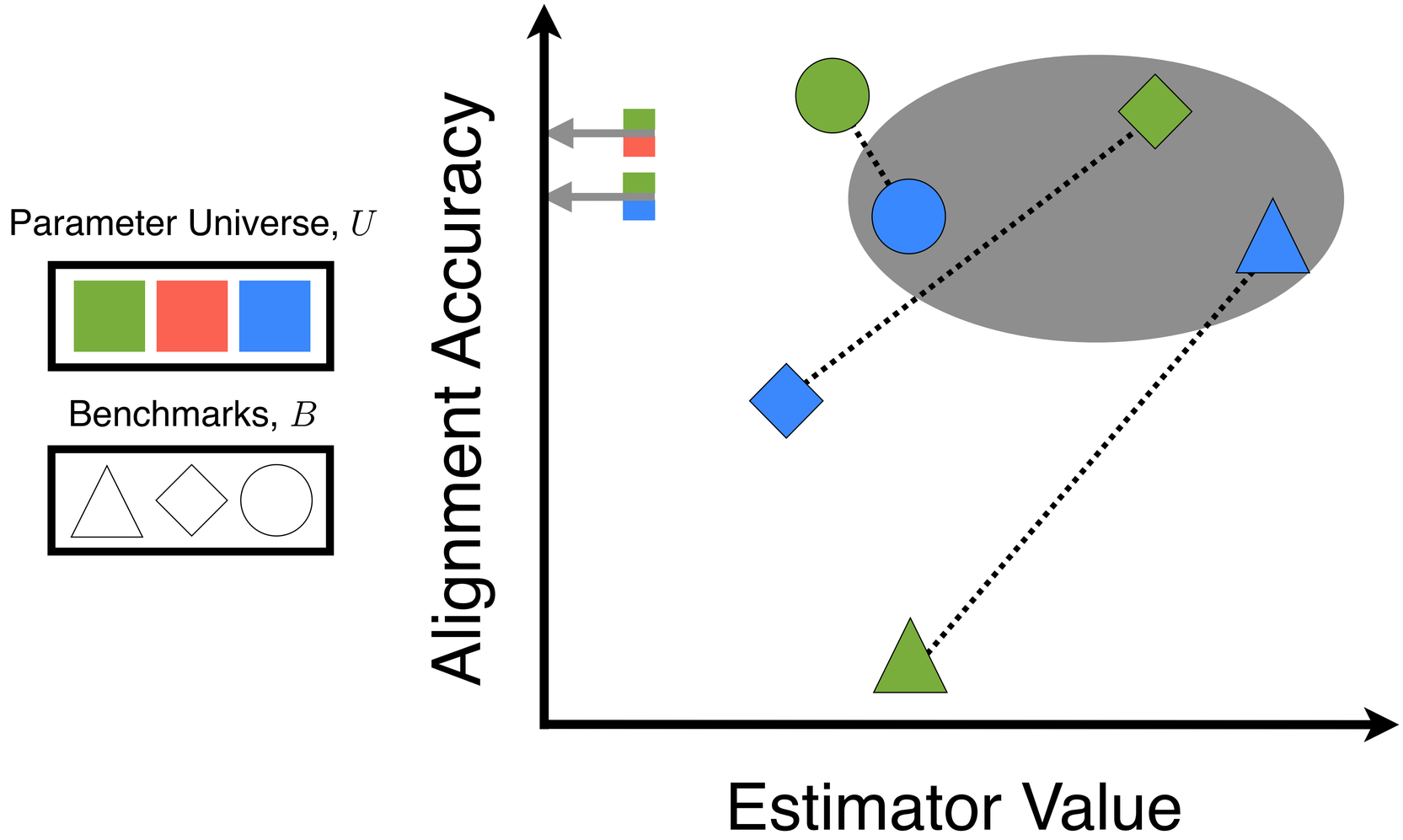
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



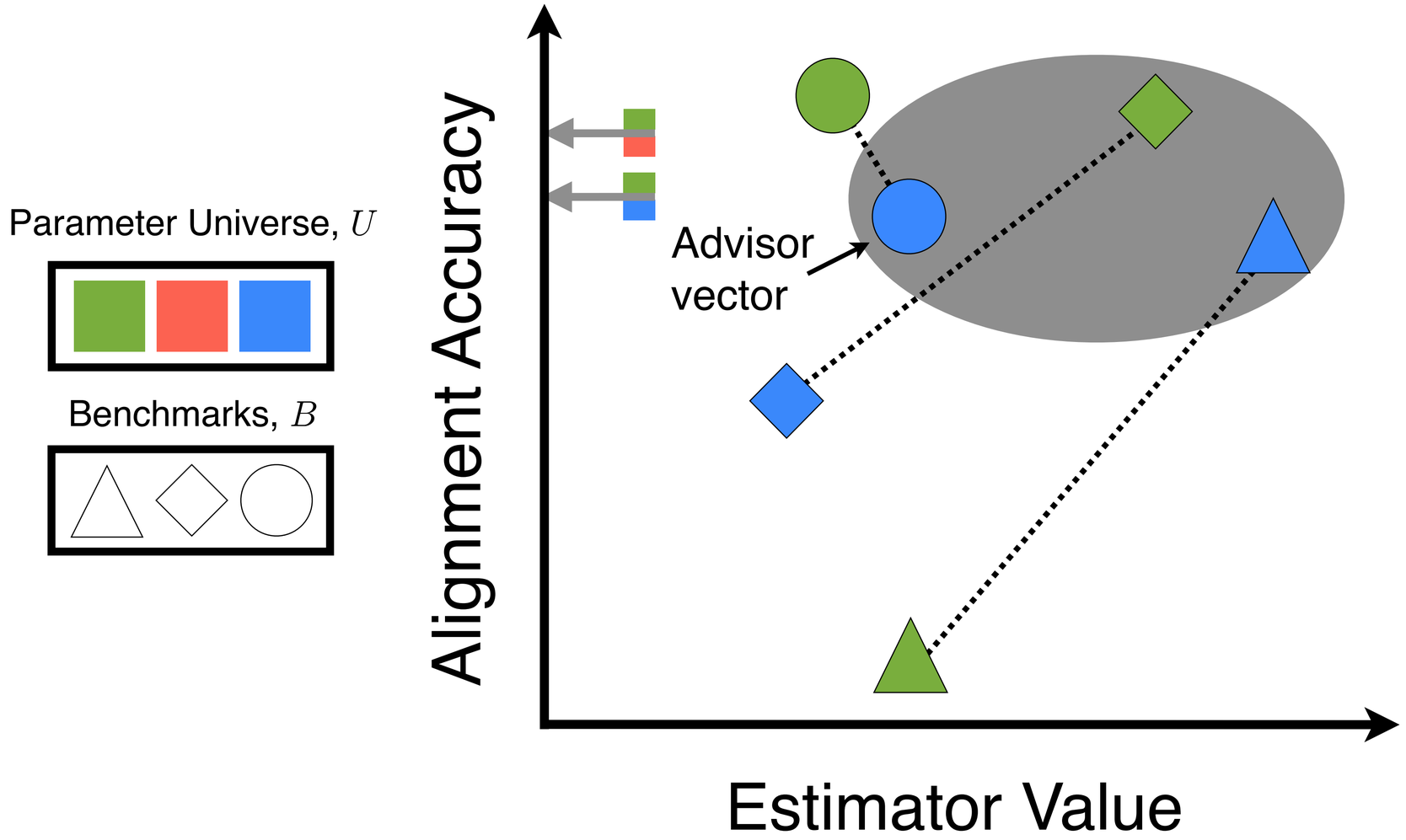
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



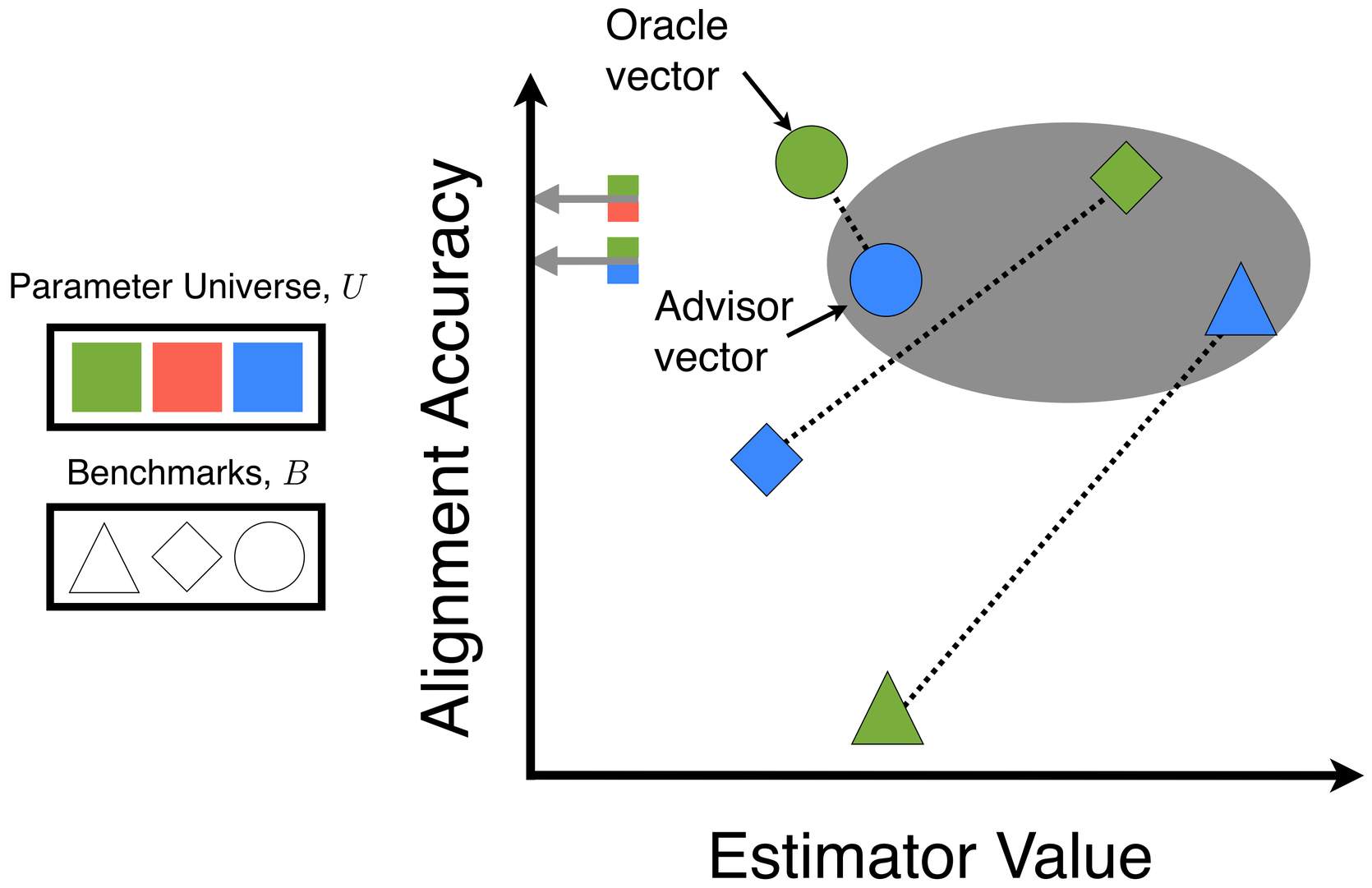
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



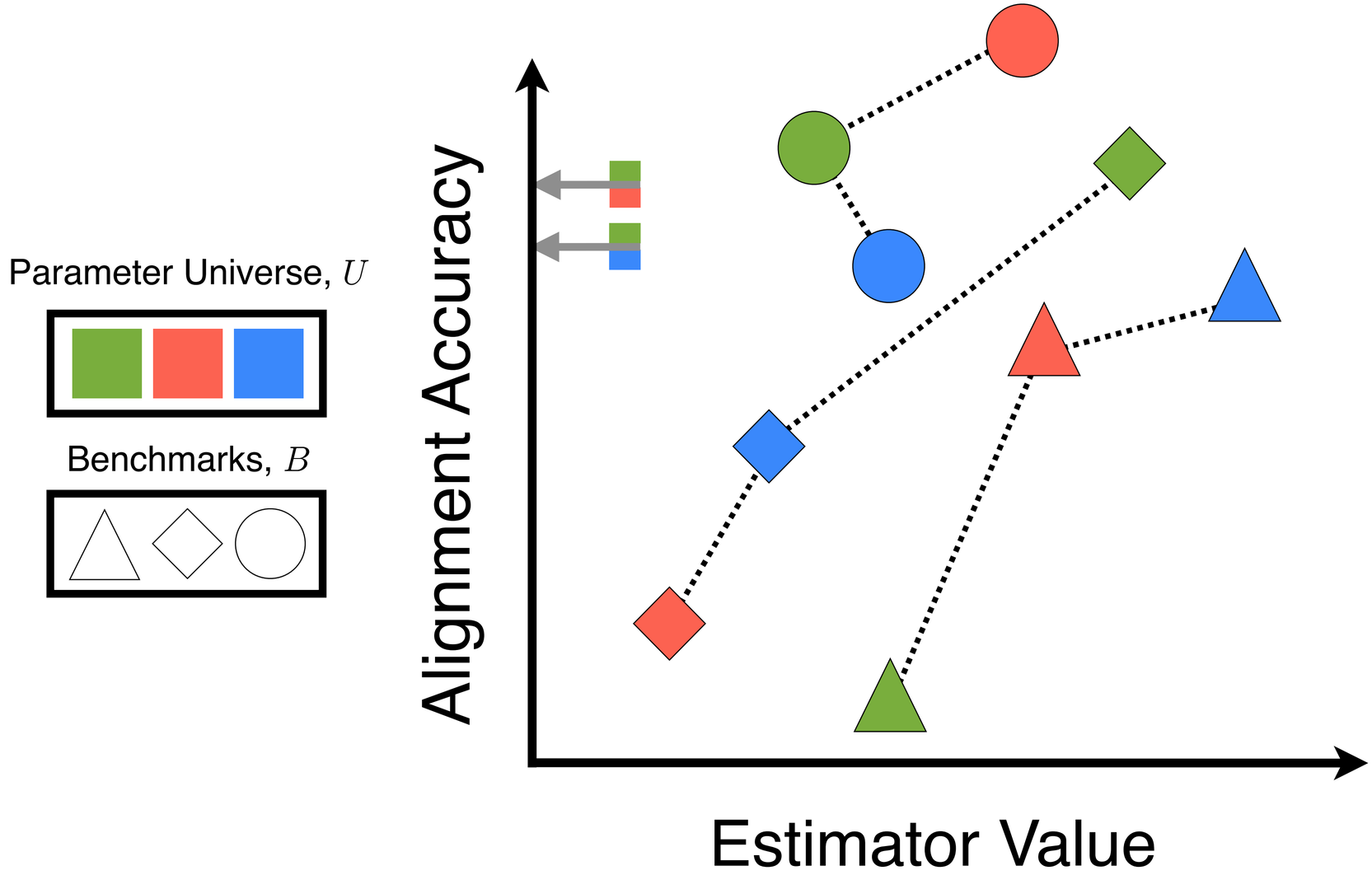
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



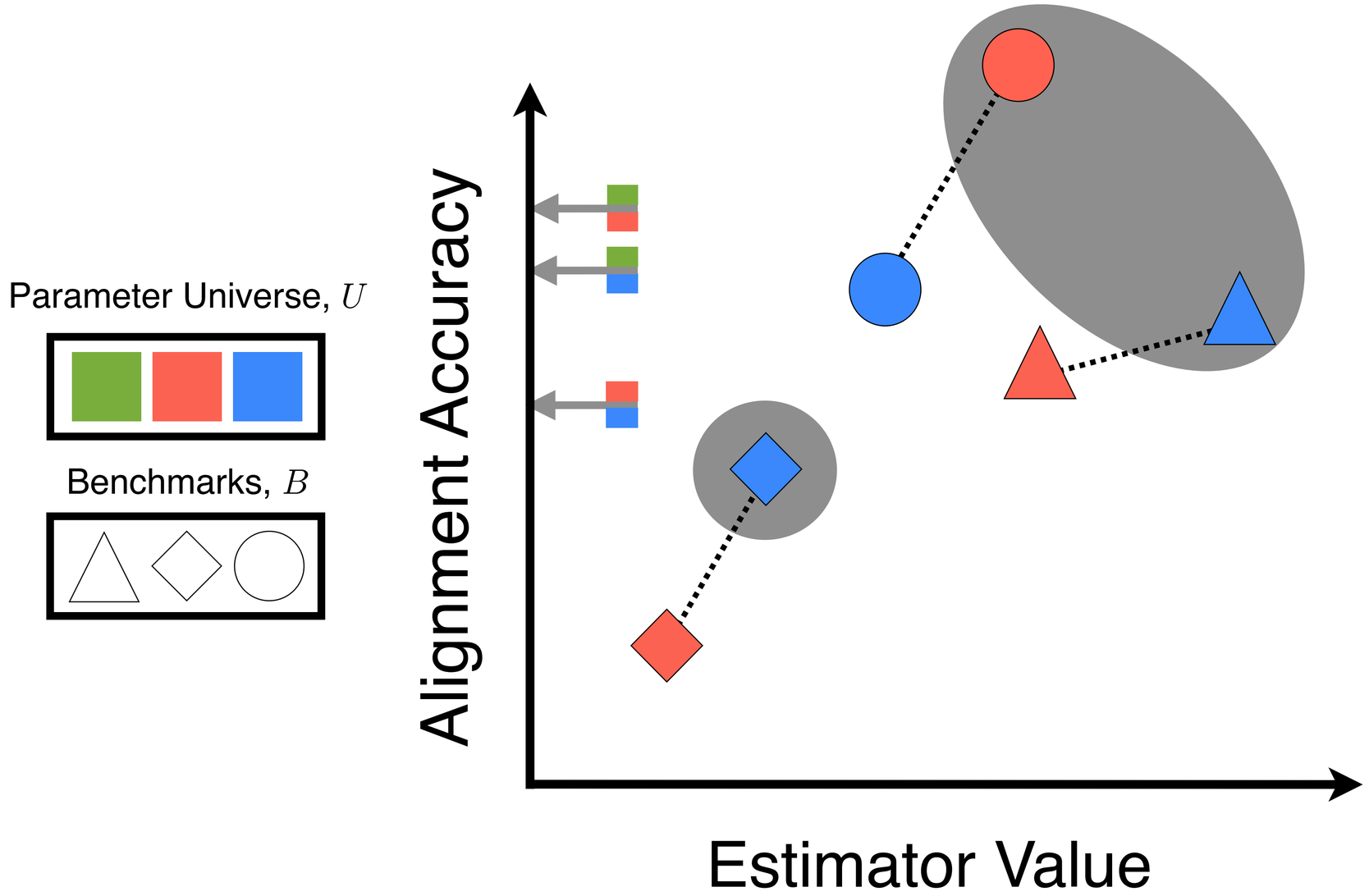
[DK ACM-BCB 2014, DK TCBB 2016] 50

Advisor Set problem



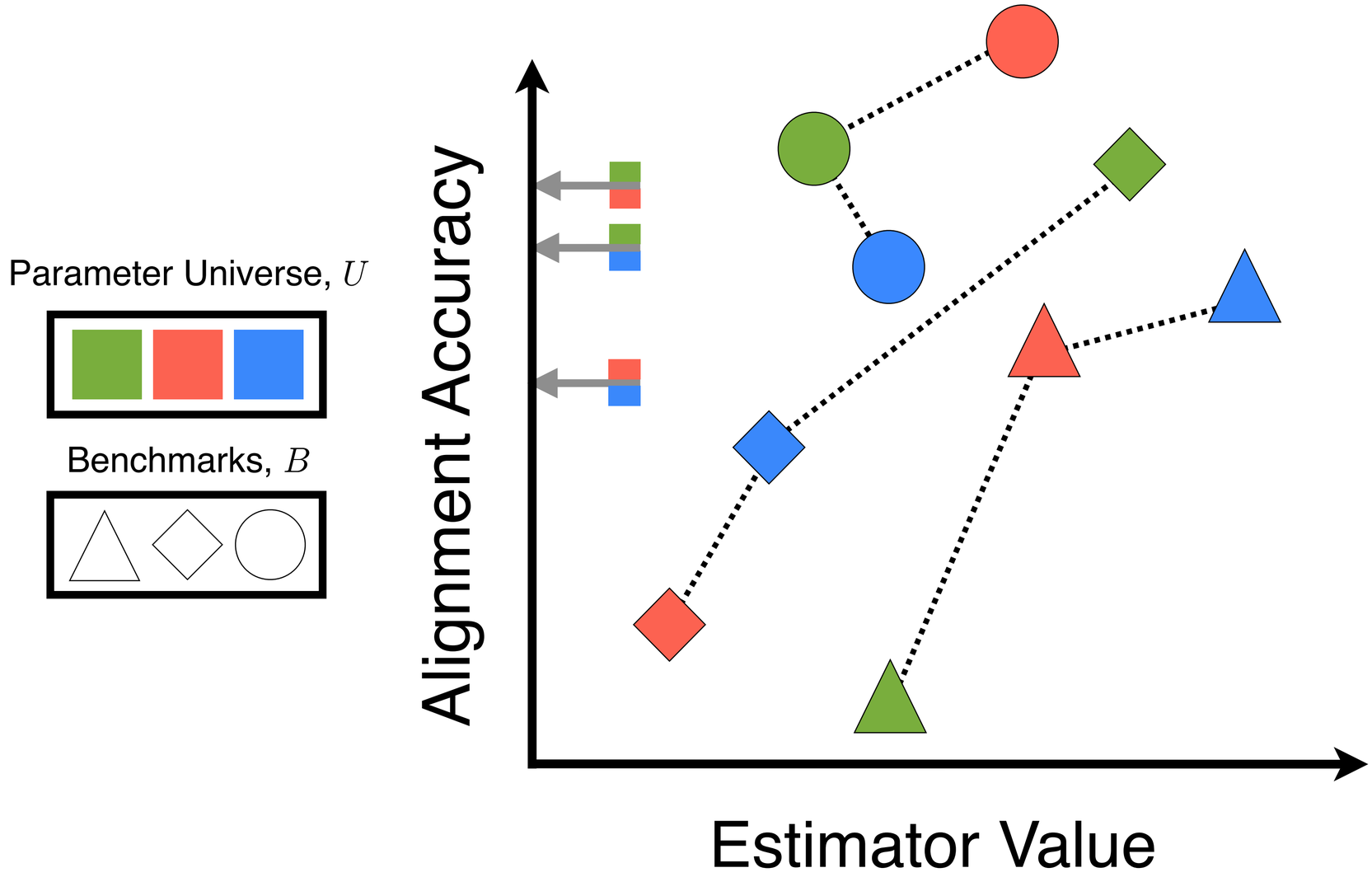
[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem



[DK ACM-BCB 2014, DK TCBB 2016]

Advisor Set problem

For the Advisor Set problem the input is

- cardinality bound k,
- universe of parameters vectors U,
- a collection of examples, each with a
  - true accuracy,
  - estimator value, and
  - benchmark weight.

Advisor Set problem

The output is

- an optimal set $P \subseteq U$ of parameter vectors with $|P| \leq k$, that maximizes

$$\sum_{\text{benchmarks } i} w_i \ \text{Accuracy}_i(P)$$

- where $\text{Accuracy}_i(P)$ is the true accuracy for benchmark i of the example chosen by an advisor that uses advisor set P

Advisor Set problem

THEOREM (Problem Complexity)

The Advisor Set problem is NP-complete.

- Polynomial-time solvable for fixed k
- Optimal oracle sets can be found for all k in practice by integer linear programming

[DK ACM-BCB 2014, DK TCBB 2016]

Approximation algorithm

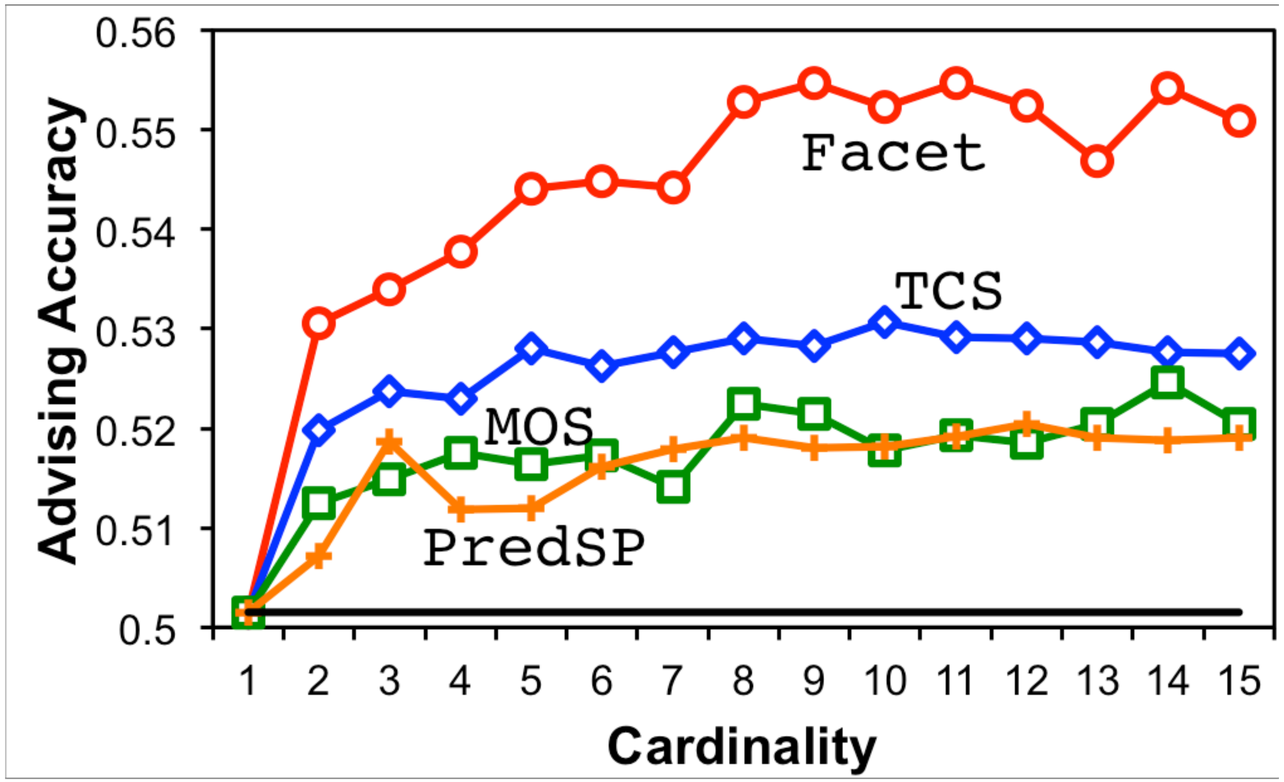
THEOREM (Approximation Algorithm)

A greedy approach yields an $\frac{\ell}{k}$-approximation algorithm for Advisor Set, for constant $\ell < k$.

The approximation ratio $\frac{\ell}{k}$ is tight.

Multiple sequence alignment advising

Advising performance for various estimators



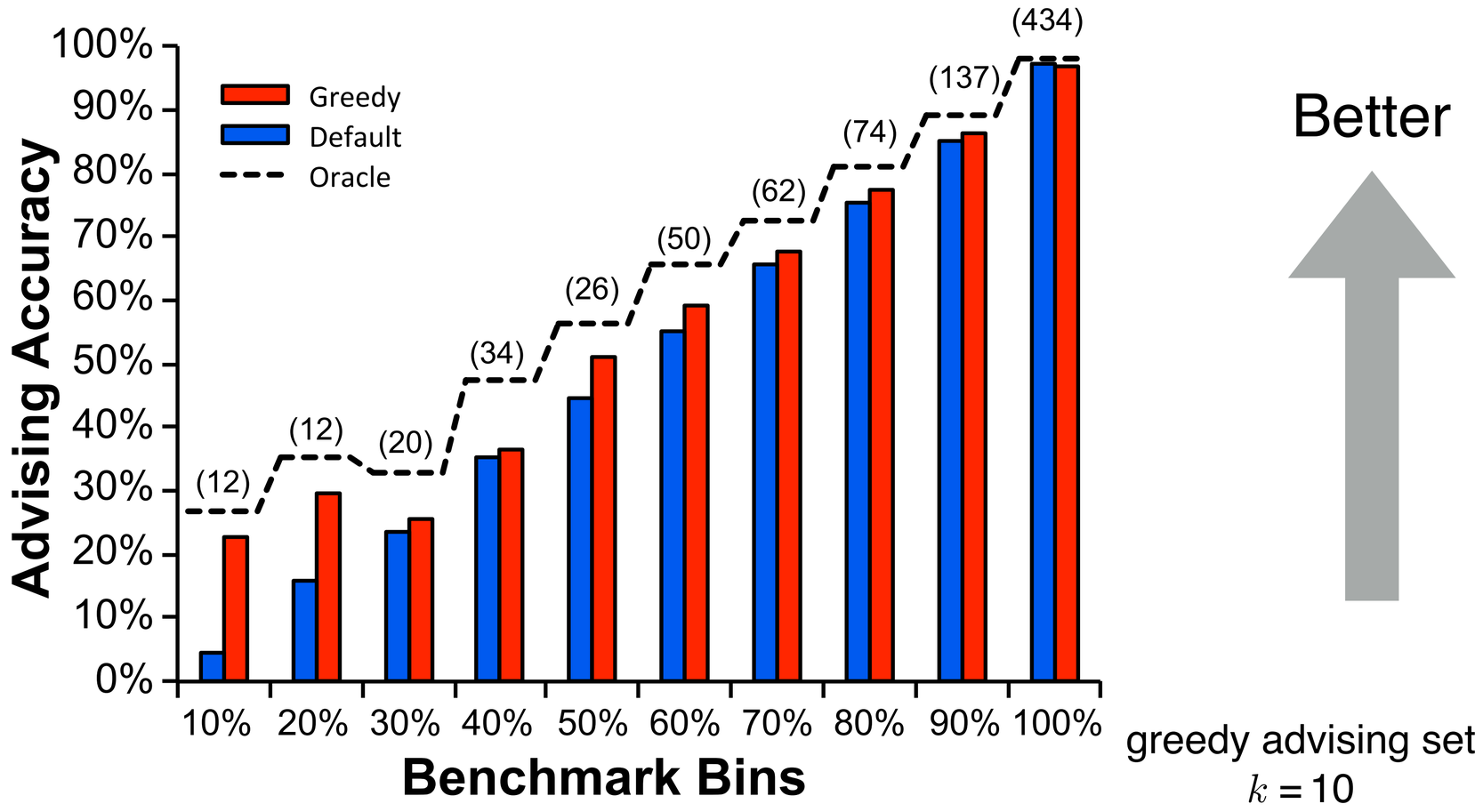
Better

For parameter advising `Facet` has the highest accuracy

[Opal: Wheeler and Kececioglu ISMB 2007]
[MOS: Lassmann and Sonnhammer, NAR 2005] [PredSP: Ahola, et al. Bioinformatics 2008] [TCS: Chang, et al. MBE 2014]

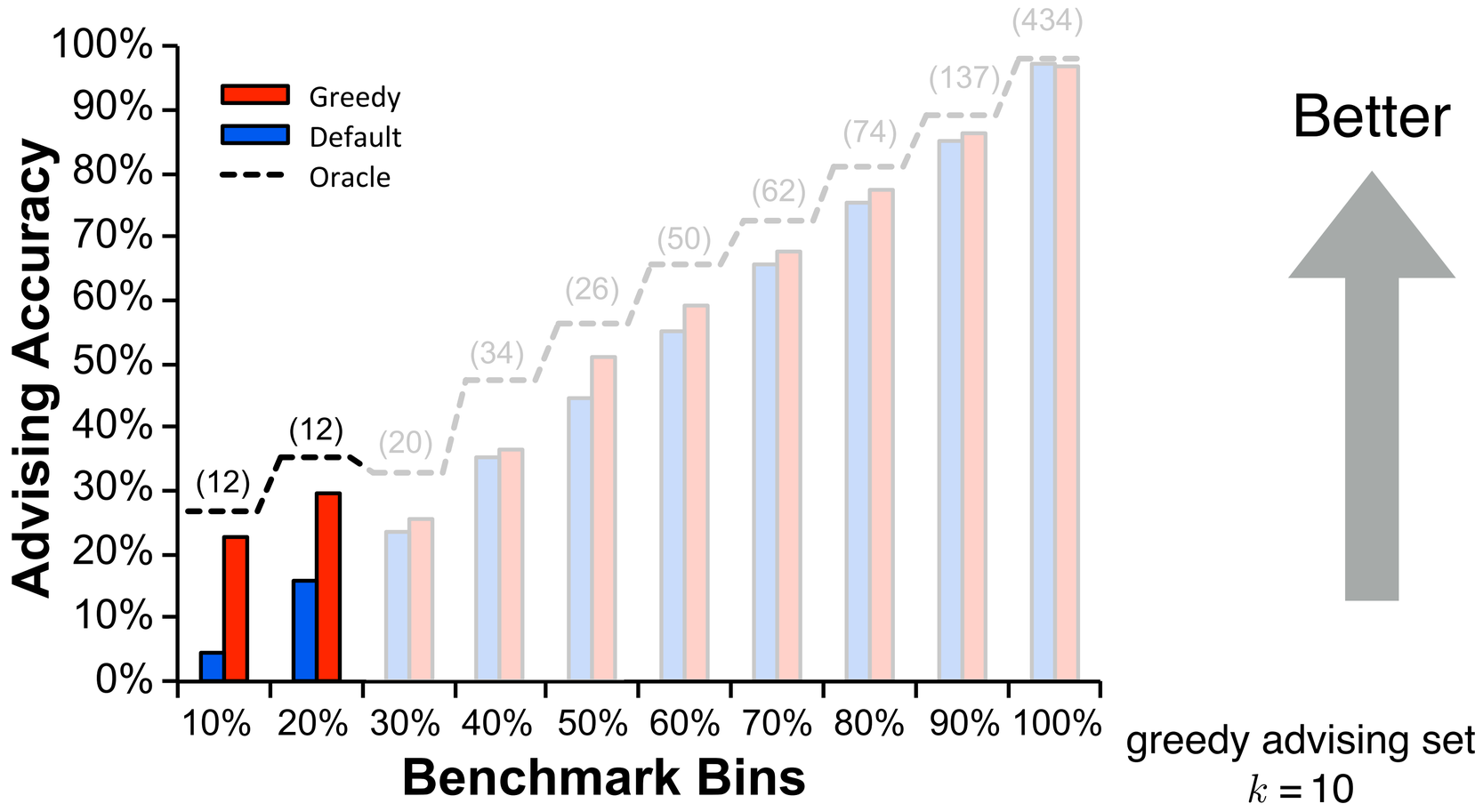
Multiple sequence alignment advising

Average accuracy of advisors by difficulty bin



Multiple sequence alignment advising

Average accuracy of advisors by difficulty bin

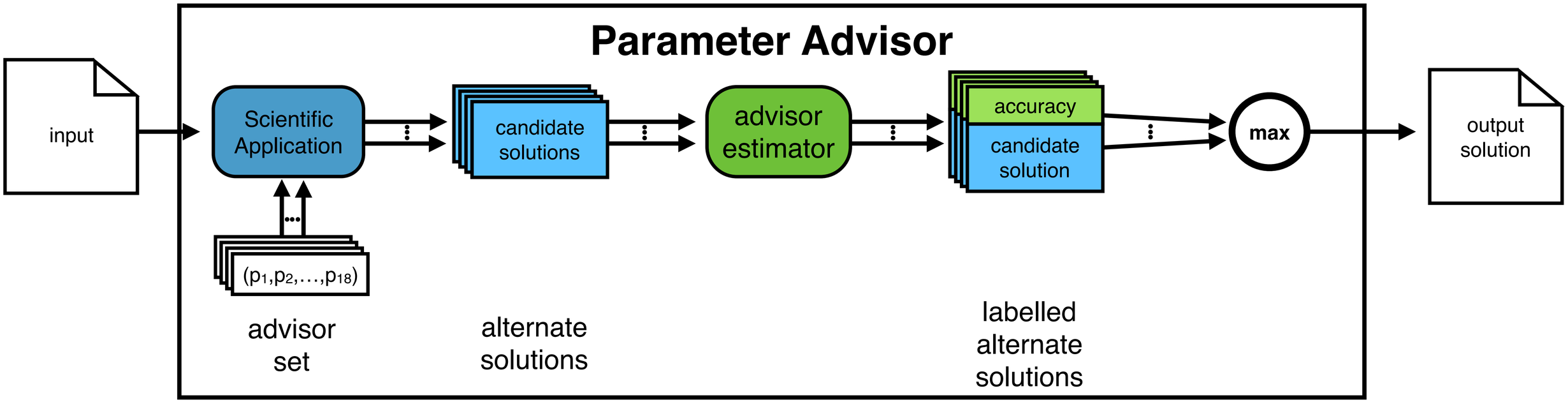


Boosts the accuracy on the hardest bins by almost 20%

Ensemble alignment

To apply parameter advising to aligner advising

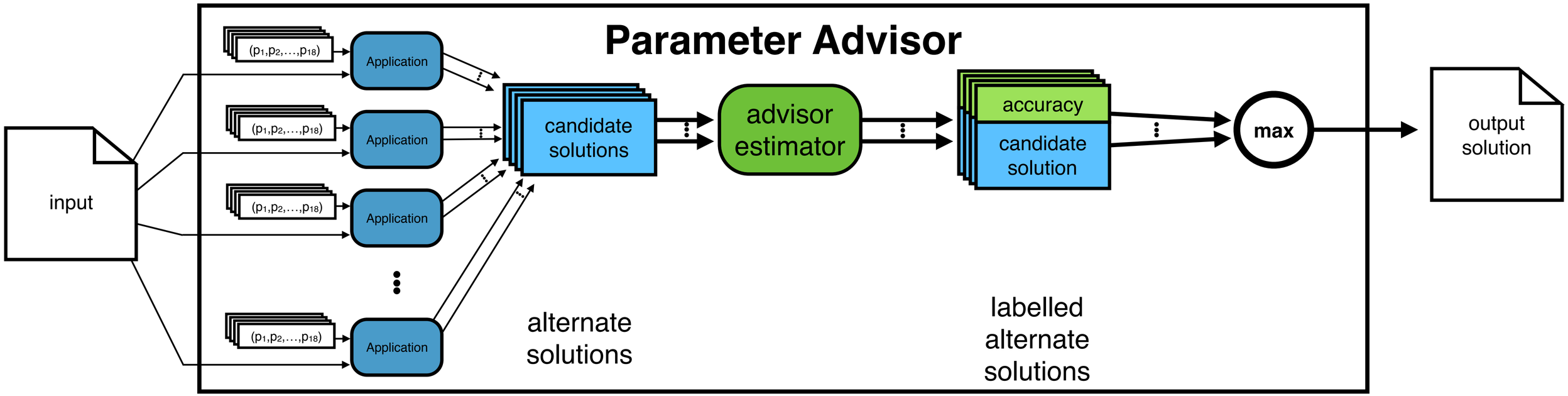
- add the choice of aligner to a parameter choice
- includes one of more choices of the tunable parameters



Ensemble alignment

To apply parameter advising to aligner advising

- add the choice of aligner to a parameter choice
- includes one of more choices of the tunable parameters



Ensemble alignment

The universe for default aligner advising includes

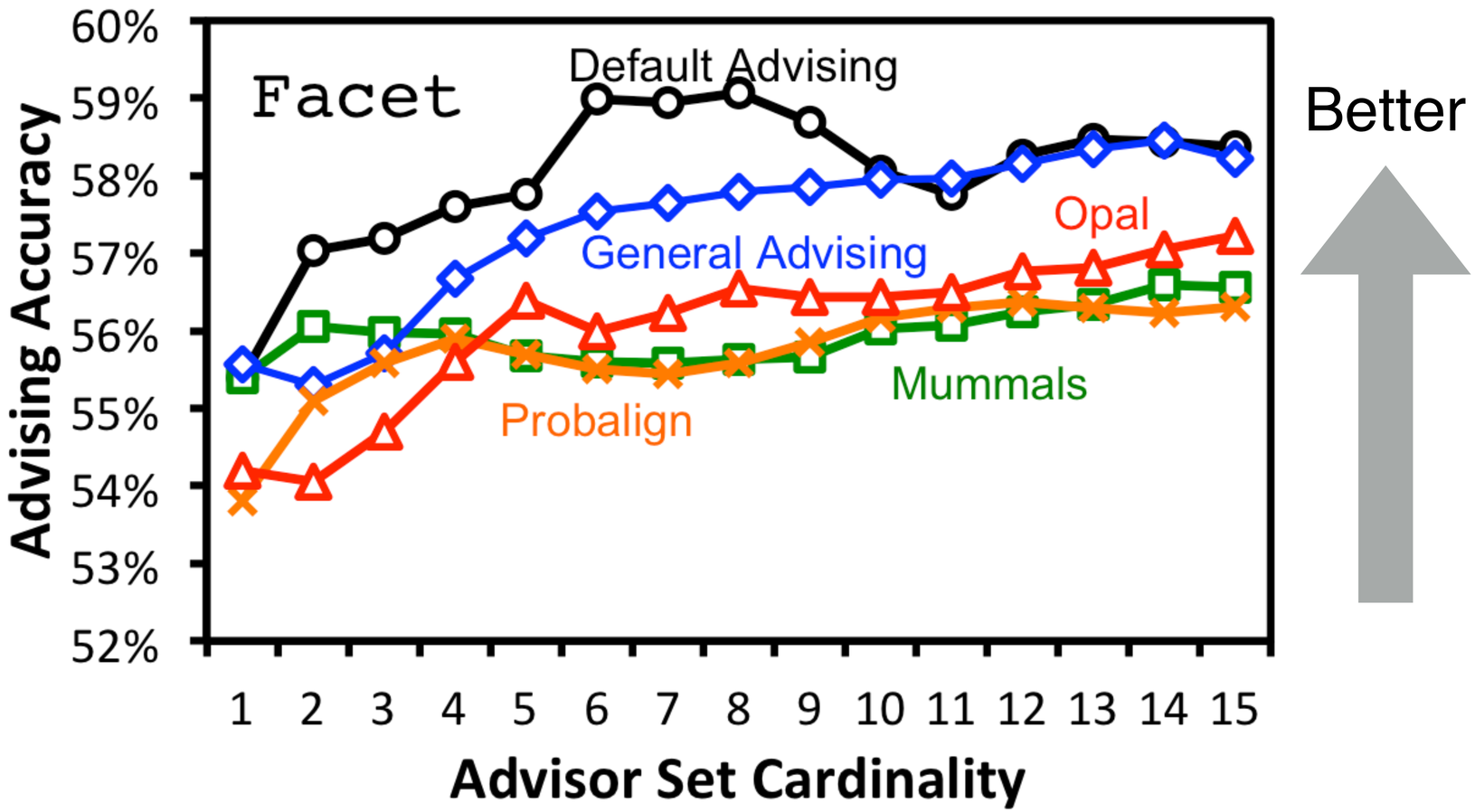
- the most commonly-used aligners (17 tools),
- using their default parameter settings.

The universe for general aligner advising adds

- a group of most-accurate aligners (10 tools)
- using expanded individual parameter universes

Ensemble alignment

Advisor performance versus set cardinality



Ensemble advising boosts accuracy over parameter advising

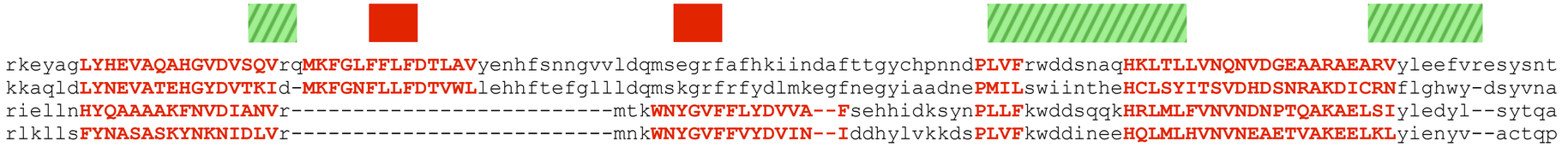
[DK ACM-BCB 2015]

Adaptive local realignment

Proteins can have different mutation rates along their length

- Alignment parameters model mutation rates
- A single choice of parameters may not be best
- Using different choices across the protein can be superior

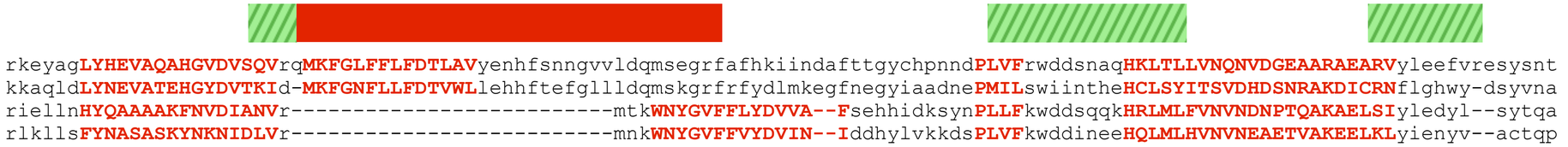
Adaptive local realignment



Adaptive local realignment

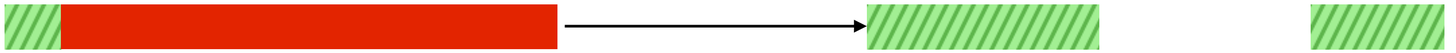
```
rkeyagLYHEVAQAHGVDVSQVrqMKFGLFFLFDTLAVyenhfsnngvvldqmsegrfafhkiindafttgychpnndPLVFrwddsnaqHKLTLLVNQNVDGEAARAEARVyleefvresysnt
kkaqldLYNEVATEHGYDVTKId-MKFGNFLLFDTVWLlehhftefgllldqmskgrfrfydlmkegfnegyiaadnePMILswiintheHCLSYITSVDHDSNRAKDICRNflghwy-dsyvna
rielln HYQAAAAKFNVDIANVr--------------------------mtkWNYGVFFLYDVVA--Fsehhidksyn PLLFkwddsqqkHRLMLFVNVNDNPTQAKAELSIyledyl--sytqa
rlklls FYNASASKYNKNIDLVr--------------------------mnkWNYGVFFVYDVIN--Iddhylvkkds PLVFkwddineeHQLMLHVNVNEAETVAKEELKLyienyv--actqp
```

Adaptive local realignment



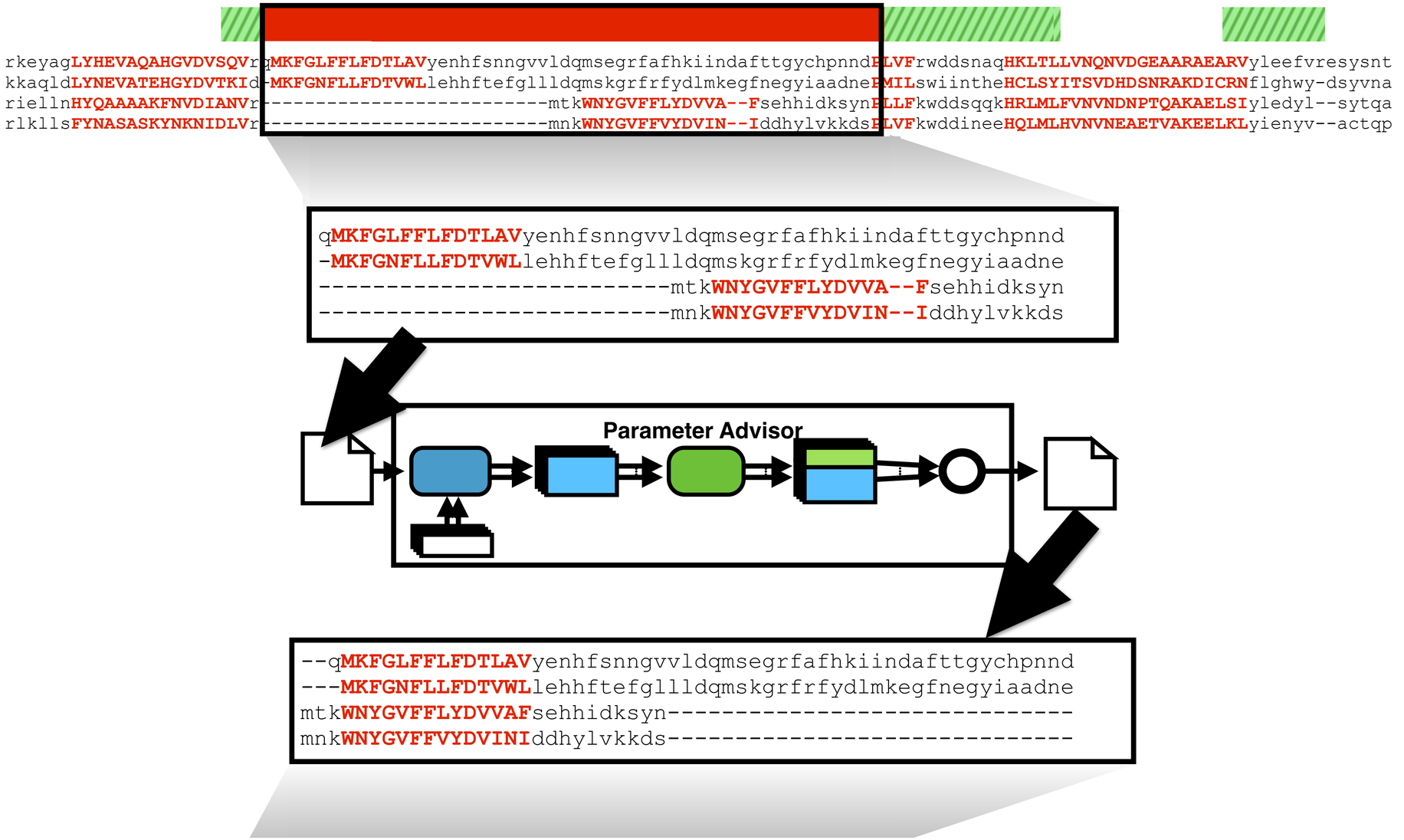
```
rkeyagLYHEVAQAHGVDVSQVrqMKFGLFFLFDTLAVyenhfsnngvvldqmsegrfafhkiindafttgychpnndPLVFrwddsnaqHKLTLLVNQNVDGEAARAEARVyleefvresysnt
kkaqldLYNEVATEHGYDVTKId-MKFGNFLLFDTVWLlehhftefgllldqmskgrfrfydlmkegfnegyiaadnePMILswiintheHCLSYITSVDHDSNRAKDICRNflghwy-dsyvna
rielln HYQAAAAKFNVDIANVr--------------------------mtkWNYGVFFLYDVVA--Fsehhidksyn PLLFkwddsqqkHRLMLFVNVNDNPTQAKAELSIyledyl--sytqa
rlklls FYNASASKYNKNIDLVr--------------------------mnkWNYGVFFVYDVIN--Iddhylvkkds PLVFkwddineeHQLMLHVNVNEAETVAKEELKLyienyv--actqp
```

Adaptive local realignment



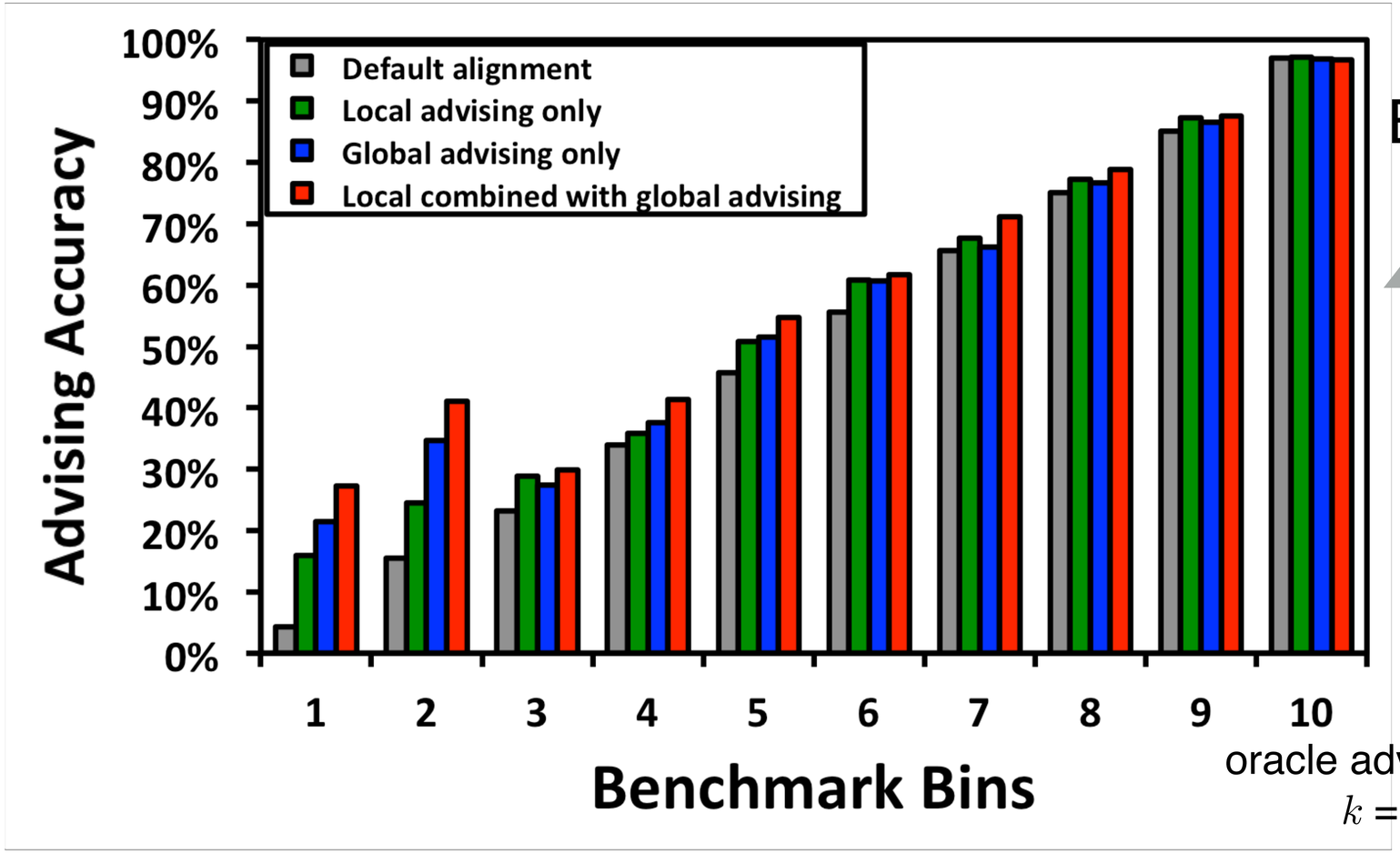
```
rkeyagLYHEVAQAHGVDVSQVrqMKFGLFFLFDTLAVyenhfsnngvvldqmsegrfafhkiindafttgychpnndPLVFrwddsnaqHKLTLLVNQNVDGEAARAEARVyleefvresysnt
kkaqldLYNEVATEHGYDVTKId-MKFGNFLLFDTVWLlehhftefgllldqmskgrfrfydlmkegfnegyiaadnePMILswiintheHCLSYITSVDHDSNRAKDICRNflghwy-dsyvna
rielnnHYQAAAAKFNVDIANVr--------------------------mtkWNYGVFFLYDVVA--Fsehhidksyn PLLFkwddsqqkHRLMLFVNVNDNPTQAKAELSIyledyl--sytqa
rlklls FYNASASKYNKNIDLVr--------------------------mnkWNYGVFFVYDVIN--Iddhylvkkds PLVFkwddineeHQLMLHVNVNEAETVAKEELKLyienyv--actqp
```

Adaptive local realignment



Adaptive local realignment

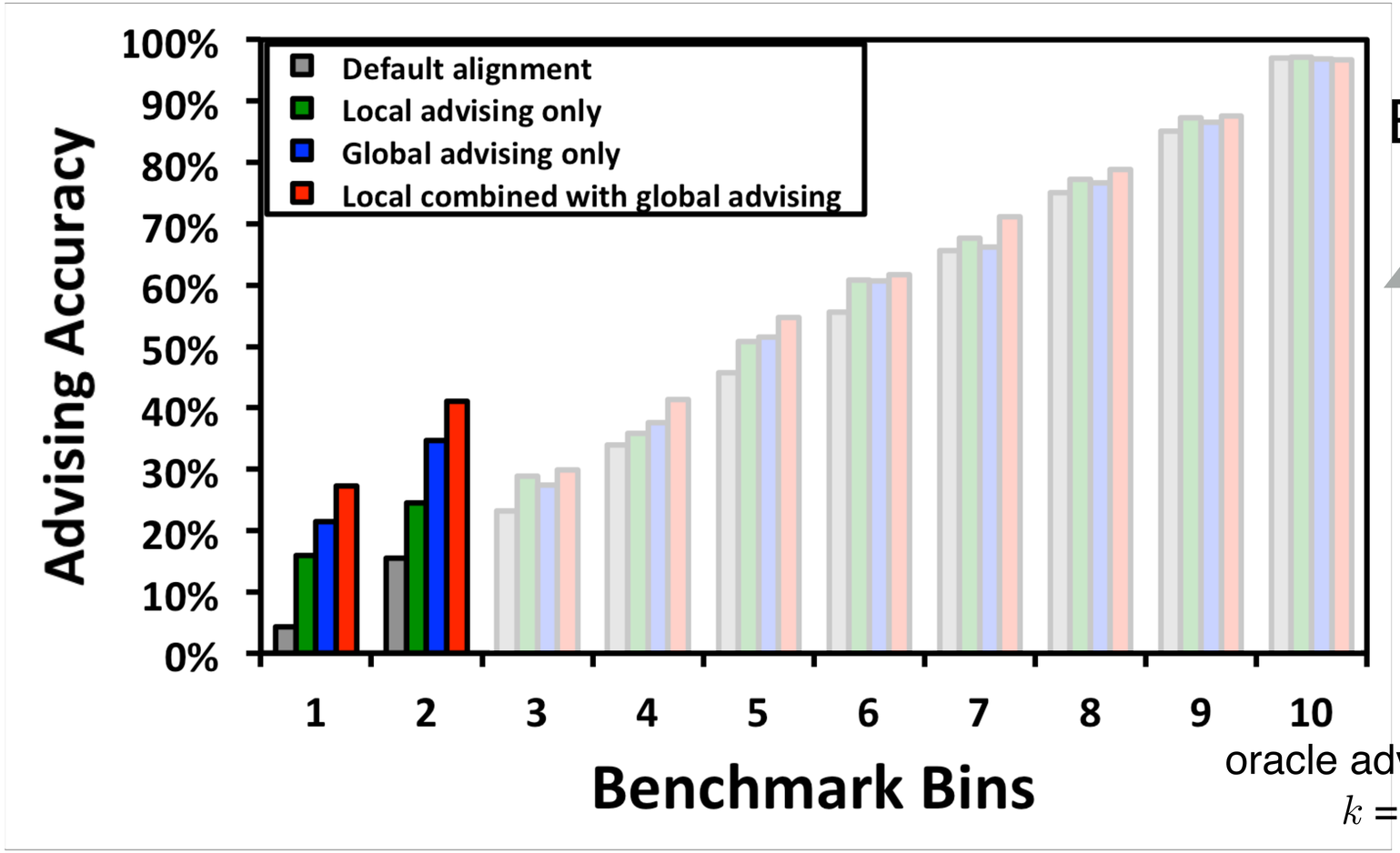
Average accuracy of advisors by difficulty bin



[DK RECOMB 2017, DK JCB 2018]

Adaptive local realignment

Average accuracy of advisors by difficulty bin



Boosts the accuracy on the hardest bins by almost 26%

[DK RECOMB 2017, DK JCB 2018]

Multiple sequence alignment advising

Parameter advising using `Facet` significantly improves accuracy

- Estimator coefficients found using linear programming
- Greedy sets are found using $\frac{\ell}{k}$-approximation algorithm
- Global advising boosts accuracy by almost 20% on hard benchmarks

Ensemble alignment further improves accuracy

- Aligner advising yields first successful ensemble aligner
- Using ensemble alignment boosts accuracy by 9% over the default

Adaptive local realignment yields further improvements

- Uses diverse parameter choices across a protein
- With global boosts accuracy by 26% over default on hard benchmarks

Other projects

Locality sensitive hashing for edit distance

- Developed a new sketching method that correlates better with edit distance than Jaccard similarity when comparing long sequences. [MDPK bioRxiv, *under review*]

Asymptotically optimal minimizers schemes

- Minimizers scheme quality depends on the ordering of k-mers used, we prove the limits and a method to construct optimal schemes. [MDK ISMB/Bioinformatics 2018]

High-thoughput phylogeny analysis

- Software (SICLE) to quickly find structural relationships by identifying summaries of sister subtrees to subtrees of interest. [DW PeerJ 2016]

Automated regular expression generation

- Given a set of positive and negative strings, can we determine the "best" regular expression to accept the positives and reject the negatives

What's next

Advising for new applications:

- Extending Facet to DNA and RNA multiple sequence alignments.
- Applying Adaptive Local Realignment to genome alignment.
- Less effort as we have a collection of methods for construction.

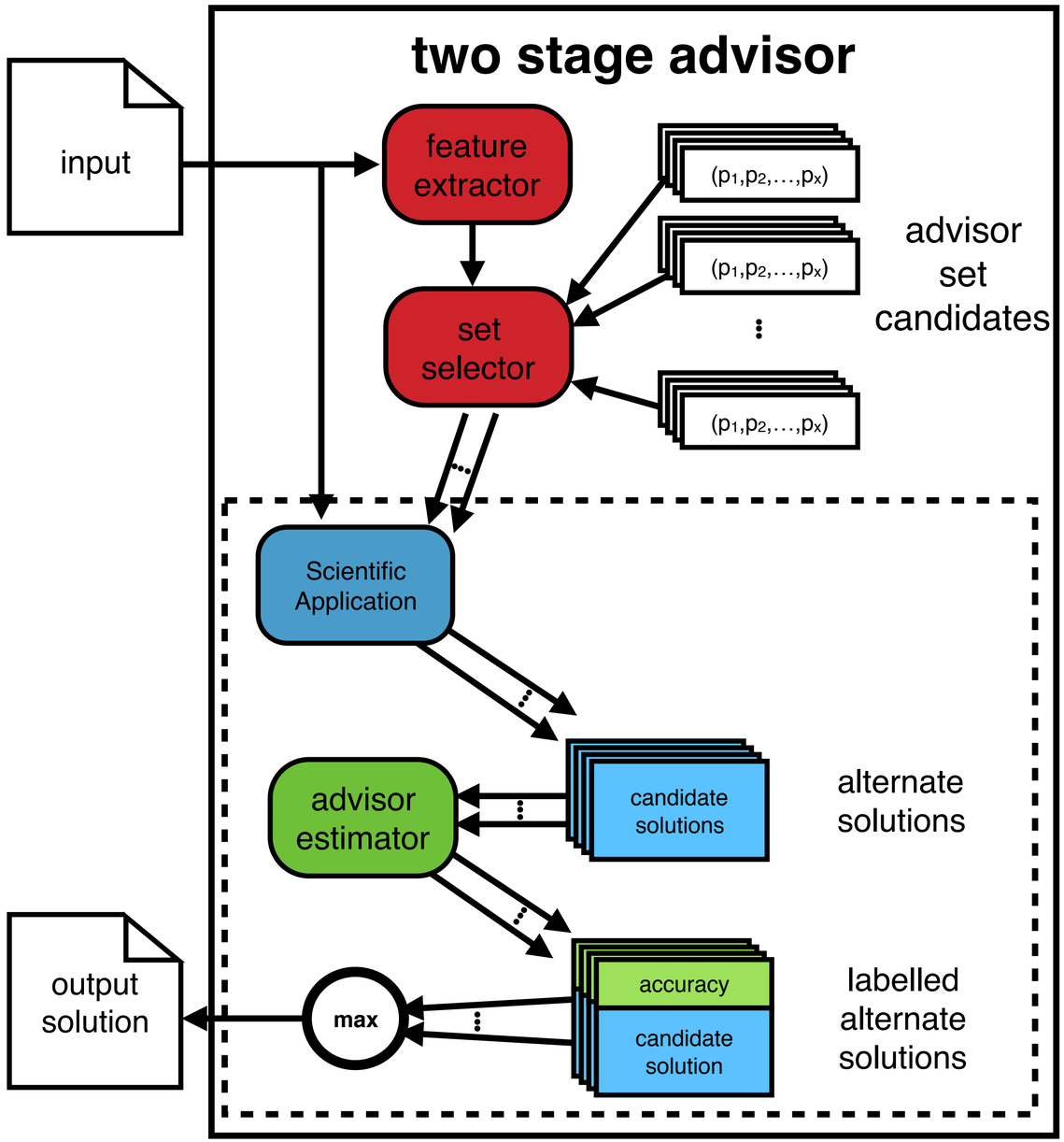
Provable limits and guarantees:

- Guarantees on the advisor's outputs distance to optimal.
- Decide if we need to run iterative optimization if room for improvement.

What's next

Combine the *a posteriori* advisor with *a priori* preprocessing.

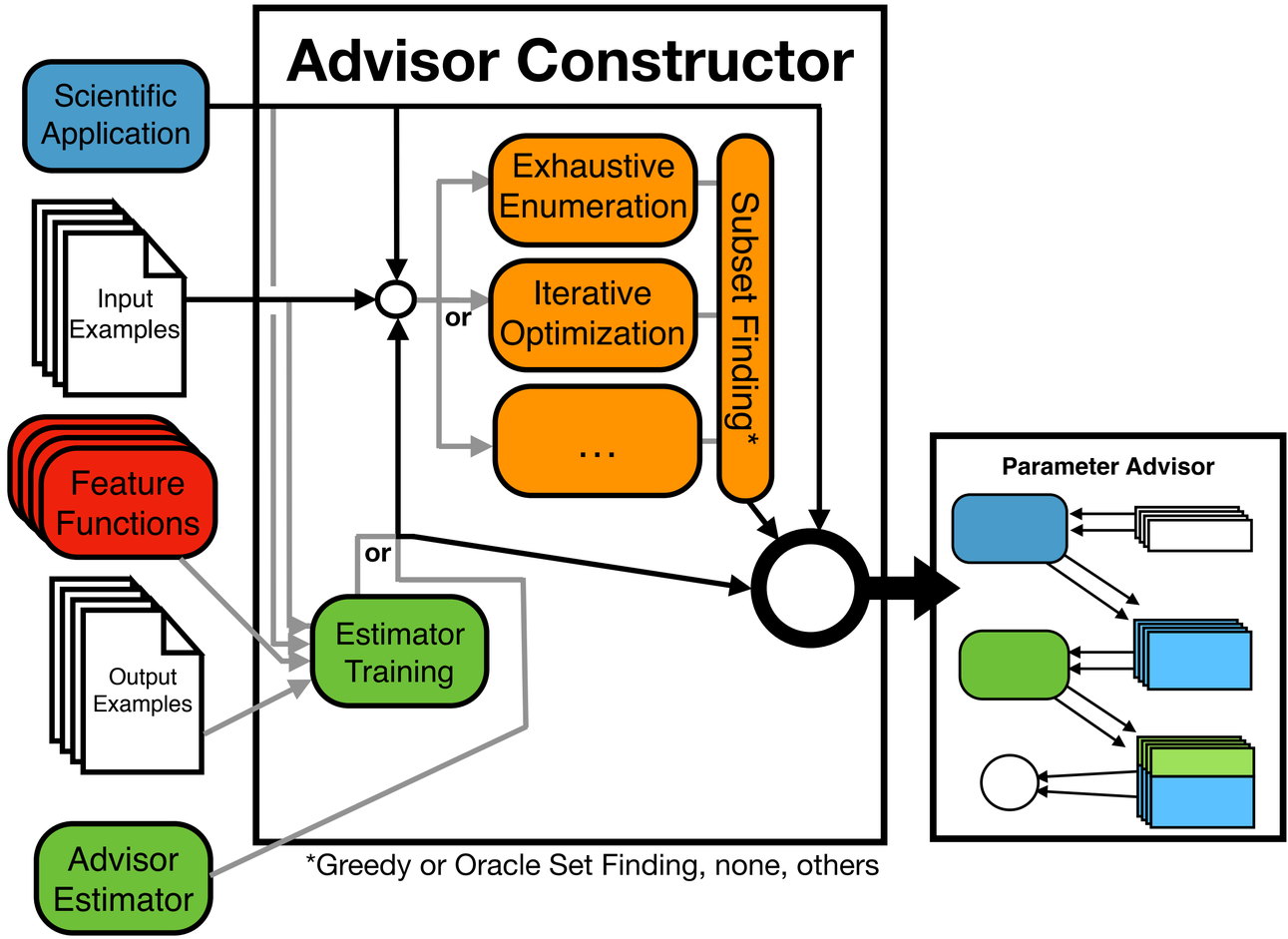
- More specialized, smaller advisor sets.



What's next

Automated advisor construction:

- Constructing a new advisor can be done programmatically.
- Draw on existing methods for constructing the two components.



Automating scientific discovery

Parameter advising is a first step to eliminating parameter choice induced errors in science.

- Improves accuracy of applications compared to default parameter choices.
- Can be applied to areas with and without standard measurements of quality.
- Unbiased way to compare new algorithms.

Acknowledgments

Software

`facet.cs.arizona.edu`

`github.com/Kingsford-Group/scallopadvising`

Contact

dandeblasio.com

danfdeblasio

Collaborators

John Kececioglu

Travis Wheeler (Montana)

Jen Wisecaver (Purdue)

Carl Kingsford

Kwanho Kim

Jonathan King

Guillaume Marçais

Prashant Pandey

Funding

CMU CBD Lane Fellowship

US National Science Foundation

US National Institutes of Health

Shurl and Kay Curci Foundation

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

Eric and Wendy Schmidt by recommendation of the Schmidt Futures program