Estimating the accuracy of multiple alignments and its use in parameter advising

Dan DeBlasio
Travis Wheeler
John Kececioglu

Department of Computer Science, University of Arizona
Janelia Farm Research Campus, Howard Hughes Medical Institute

Motivation

Estimating alignment accuracy without a reference is an important problem.

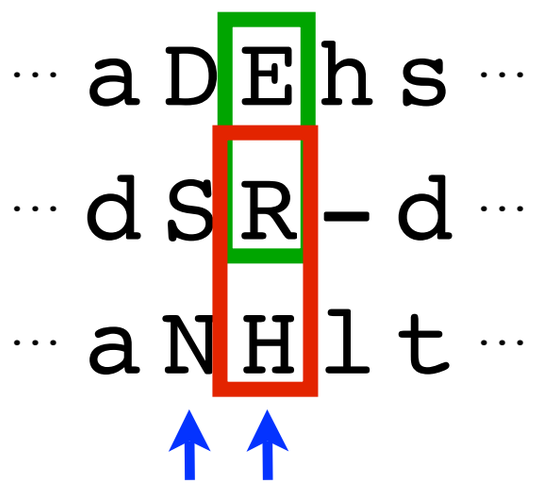
Directly applicable to

- choosing aligners for given input sequences,
- choosing parameters for a given aligner.

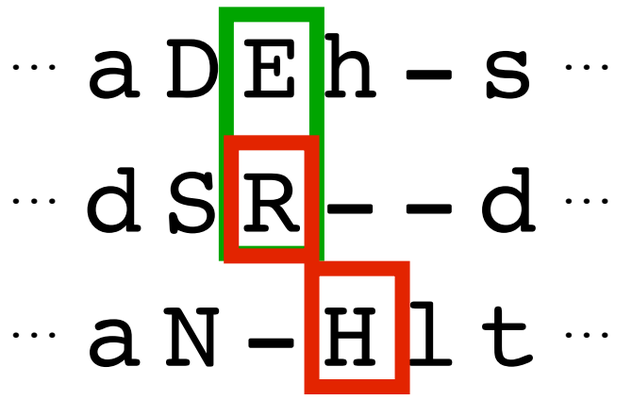
Motivation

Alignment accuracy is measured with respect to a reference alignment.

reference alignment



computed alignment



66.7% accuracy

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

Related work

Scoring-function-based approaches convert local features of an alignment A into an overall score.

- `Al2Co` [Pei and Grishin 2001]: conservation-based
- `NorMD` [Thompson *et al.* 2001]: normalized score
- `PredSP` [Ahola *et al.* 2008]: beta-distribution-based

Related work

Support-based approaches use a collection C of alternate alignments, and measure the agreement of A with C.

- MoS [Lassmann *et al.*, 2002]: vote on substitutions
- HoT [Landan and Grau, 2008]: reverse input sequences
- `Guidance` [Penn *et al.*, 2010]: alter guide tree
- PSAR [Kim and Ma, 2011]: resample HMM

Contributions

Our approach Facet ("Feature-based ACcuracy EsTimator")

- estimates accuracy by a polynomial on the features,
- efficiently learns the polynomial coefficients from examples,
- uses novel features that are fast to evaluate,
- utilizes an optimal feature subset.

Applied to parameter advising, Facet:

- finds an optimal parameter set of a given cardinality,
- outperforms other estimators in accuracy across the full range of benchmarks,
- boosts aligner accuracy on hard benchmarks by 20% over the best default parameter choice.

Estimator

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

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

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.

Learning the estimator

Difference-fitting tries to find a monotonic estimator that matches positive differences in true accuracy.

$$c^* := \operatorname*{argmin}_{c \in \mathcal{R}^t} \sum_{(A,B) \in \mathcal{P}} w_{AB} \left(\max \left\{ \left(F(B) - F(A)\right) - \left(E_c(B) - E_c(A)\right), 0 \right\} \right)^p$$

all possible coefficients

all important pairs of examples

true accuracy difference

estimated difference

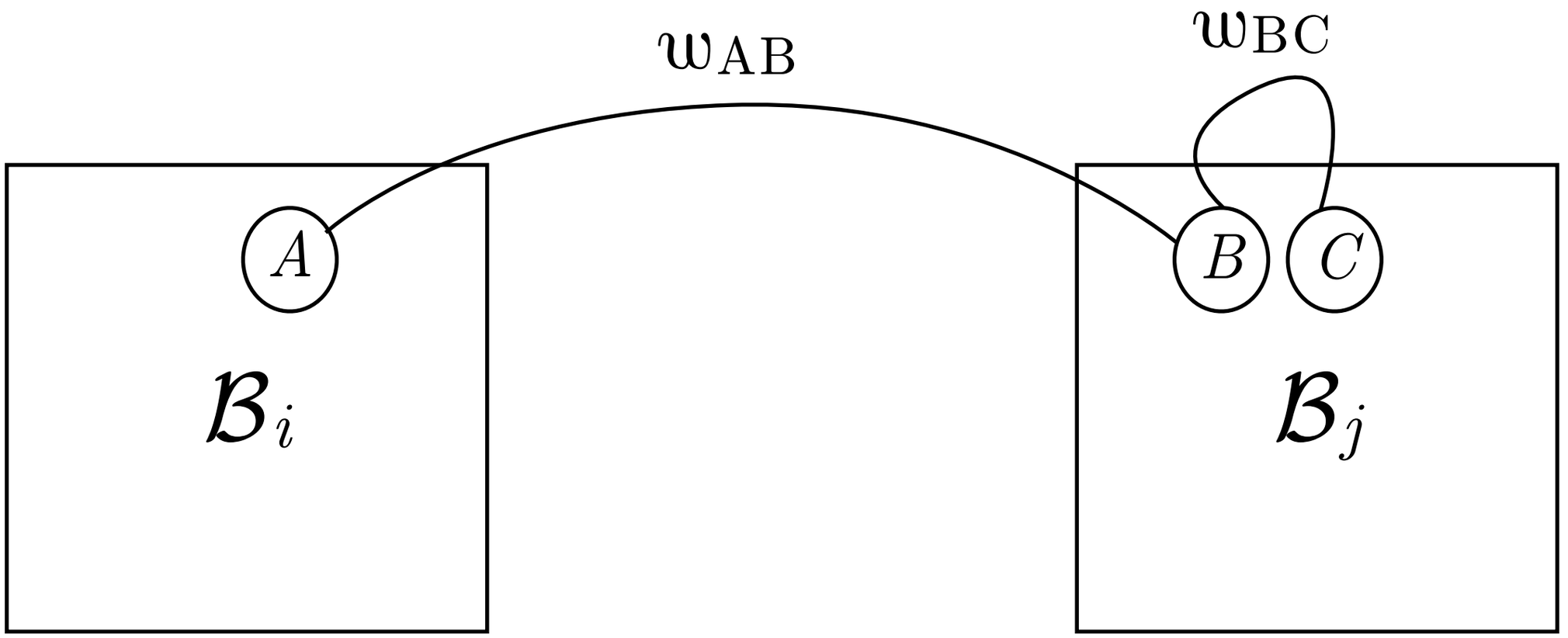
only penalize underestimating differences

controls influence of large errors

Learning the estimator

We find weights w_{AB} on pairs $(A,B) \in \mathcal{P}$ to weight bins equally.

- place $\frac{1}{2} w_{AB}$ on the bins that contain A and B,



Learning the estimator

We find weights w_{AB} on pairs $(A,B) \in \mathcal{P}$ to weight bins equally.

- place $\frac{1}{2} w_{AB}$ on the bins that contain A and B,
- each bin $\mathcal{B}$ receives total weight 1.

$$\sum_{\substack{(A,B) \in \mathcal{P} \\ A \in \mathcal{B}}} \frac{1}{2} w_{AB} + \sum_{\substack{(A,B) \in \mathcal{P} \\ B \in \mathcal{B}}} \frac{1}{2} w_{AB}$$

We call such w_{AB} balanced weights.

Learning the estimator

Theorem (Existence of Balanced Weights)

Suppose every bin $\mathcal{B}$ has some pair $(A, B) \in \mathcal{P}$ with both alignments $A, B \in \mathcal{B}$.
Then balanced weights always exist.

Theorem (Finding Balanced Weights)

When the above holds, we can find balanced weights in $O(k+m)$ time for k bins and m pairs.

Feature functions

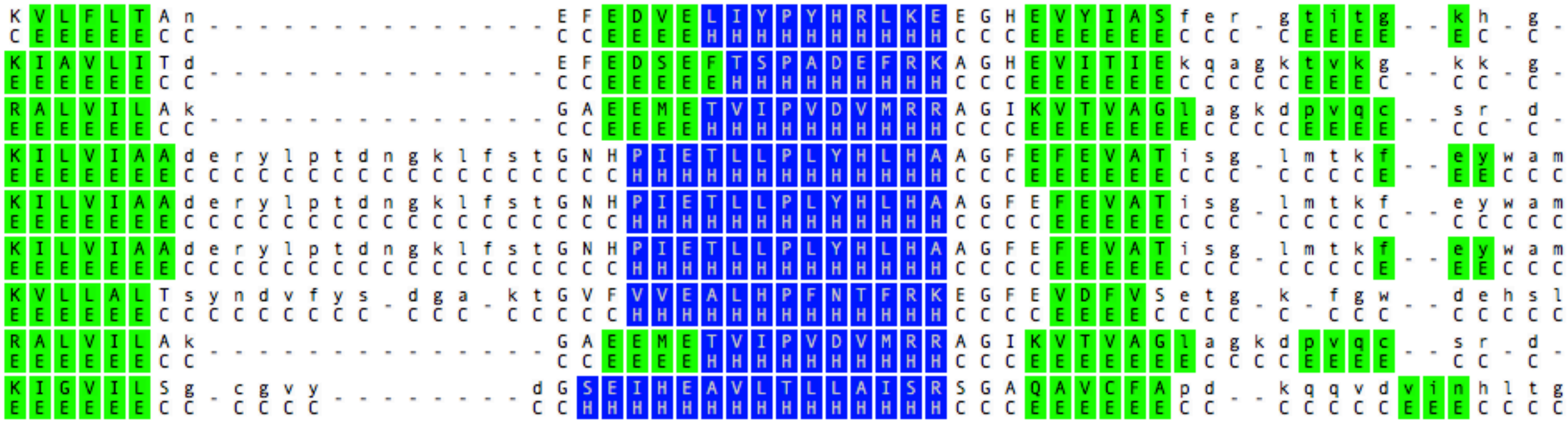
Features based only on the input alignment

- Amino Acid Identity
- Substitution Compatibility
- Gap Open Density
- ...

Features

There are three types of secondary structure

- α-helix,
- β-sheet,
- coil.



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

Feature functions

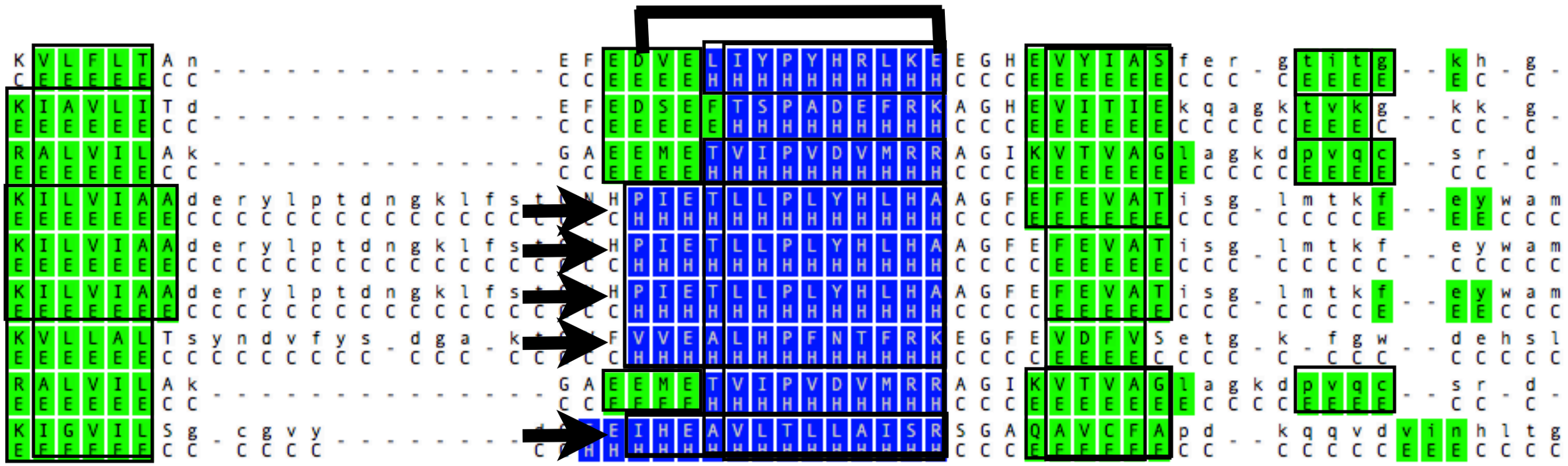
Features using predicted secondary structure

- 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.

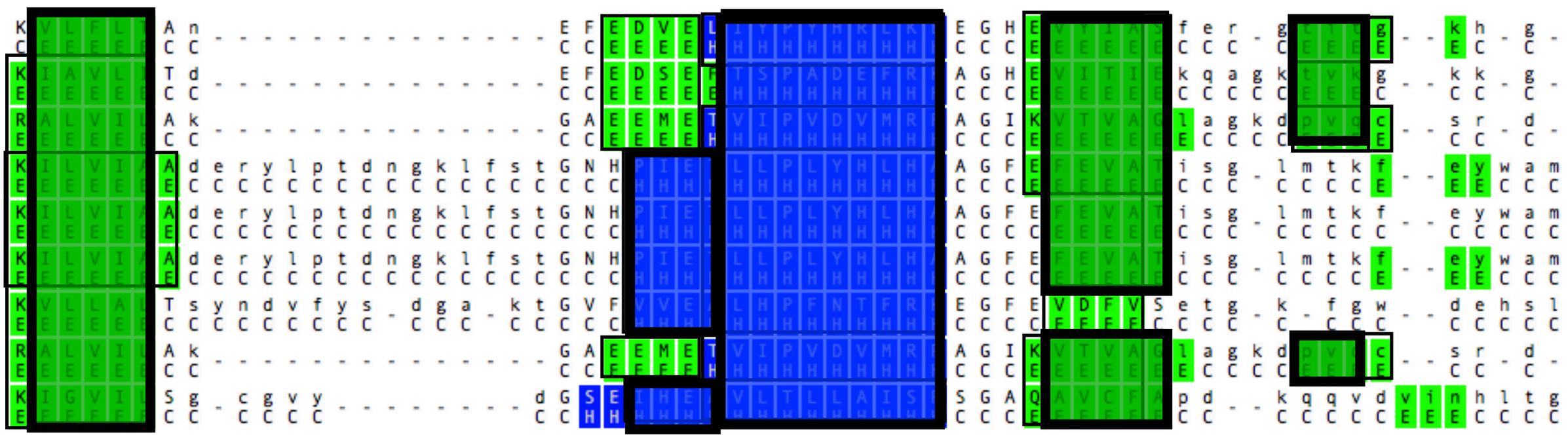


Secondary structure blockiness

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.

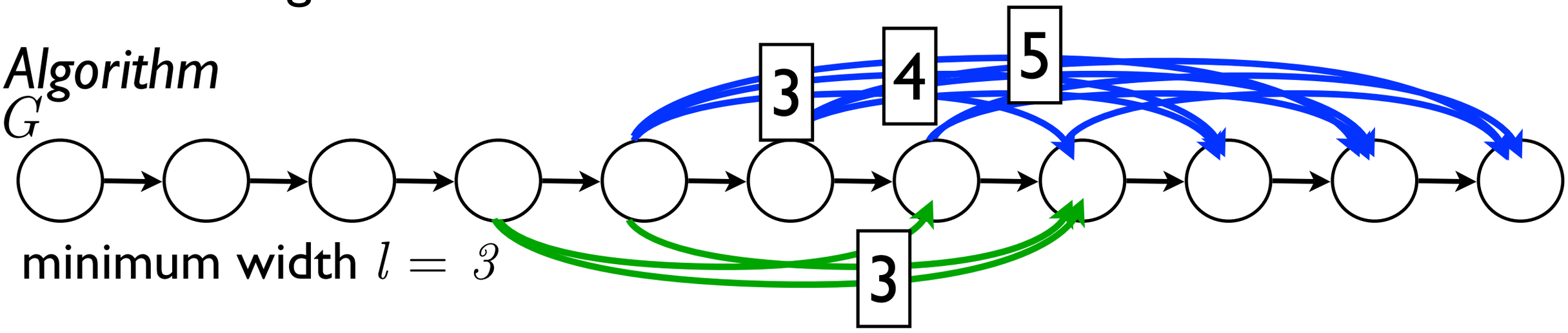


Secondary Structure Blockiness

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.

Parameter advising

Aligners often use *one* default parameter choice for *all* inputs.

- The default attempts to have good *average* accuracy across benchmarks.
- An optimal default choice can be found by inverse alignment [Kececioglu and Kim 2007].
- The default may be a poor choice for specific inputs.

Can we boost aligner accuracy
by an input-dependent choice
of parameter values?

Parameter advising

Parameter advising is selecting a parameter choice p from a set P to maximize the accuracy of an aligner $\mathcal{A}$.

- Given estimator E, an advisor finds a parameter choice $\tilde{p}$ for input sequences S.

$$\tilde{p} \quad := \quad \underset{p \in P}{\operatorname{argmax}} \; E\Big(\mathcal{A}_p(S)\Big)$$

- The oracle is a perfect advisor that uses true accuracy $F(A)$.

Parameter Advising

We want to find the best set P of k parameter choices.

- P is drawn from a universe of parameters.
- Assign each benchmark to best parameter in P.
- Select P to maximize average accuracy across benchmarks.

Finding the best P can be reduced to

- the Facility Location Problem,
- which we solve by integer linear programming.

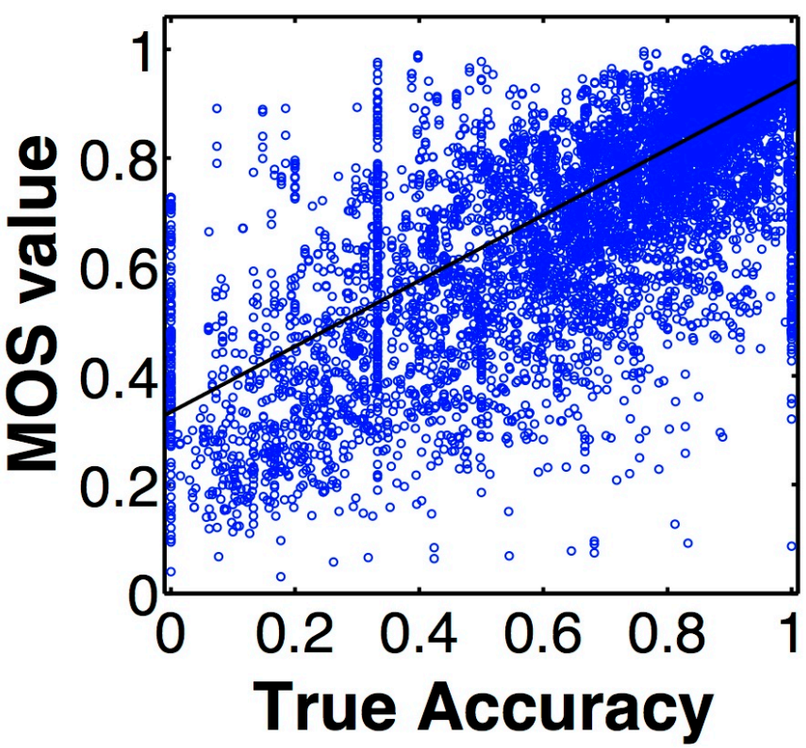
Experiments

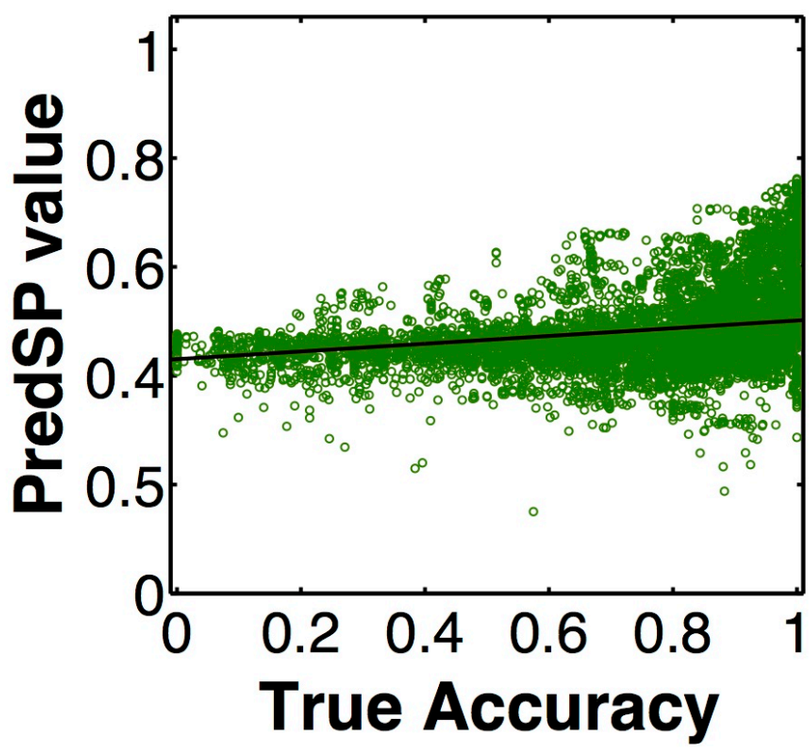
We evaluate Facet as a parameter advisor

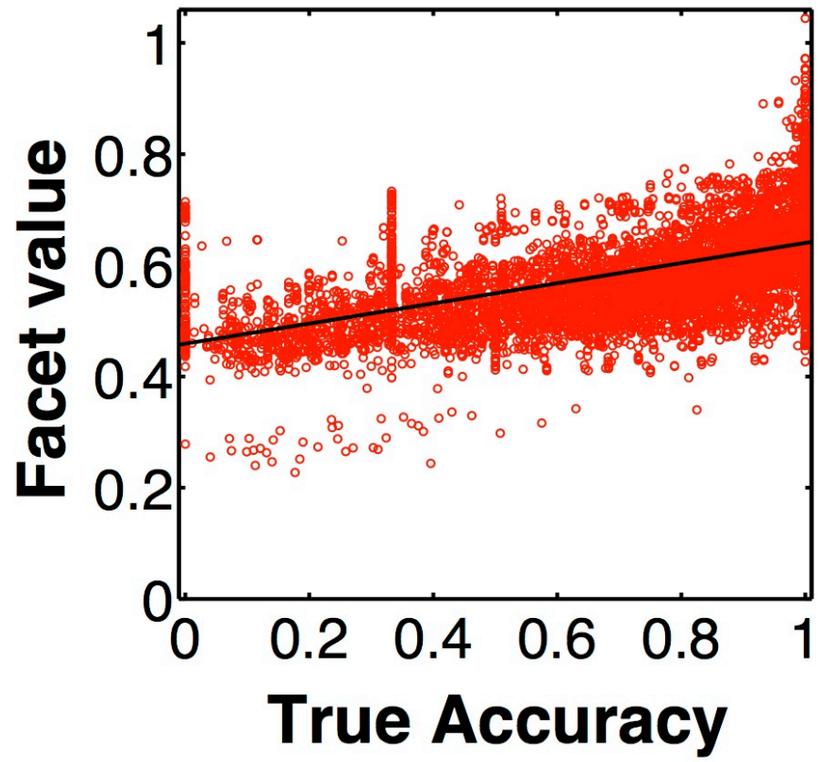
- compared against NorMD, PredSP, MoS, and HoT,
- on 800 benchmark alignments from BENCH and PALI,
- with a universe of 3200 parameter choices,
- trained and tested with 3-fold cross validation,
- advising parameter choices for the Opal aligner.

Experimental results

These estimators display very different trends.



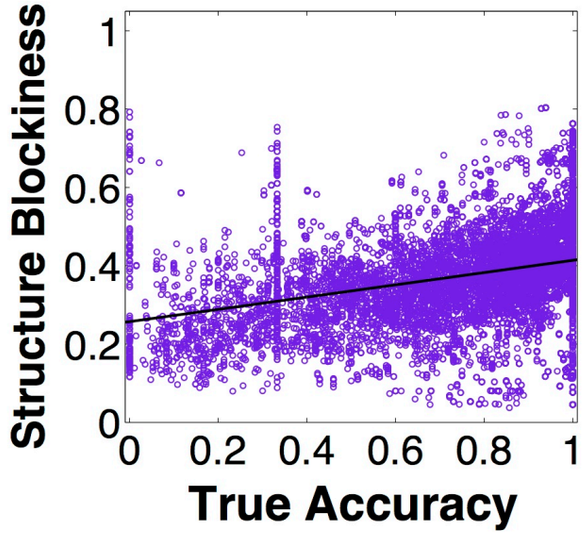


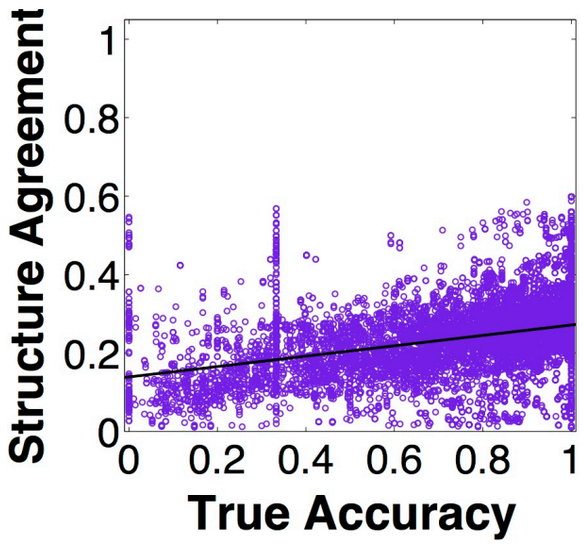


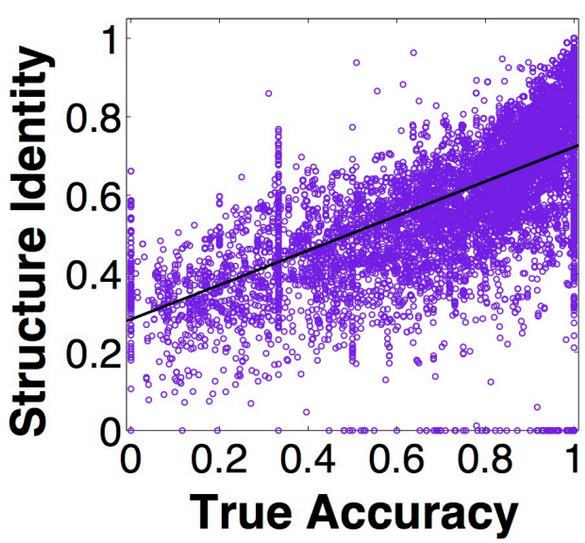
For parameter advising, an estimator needs to have good slope and spread.

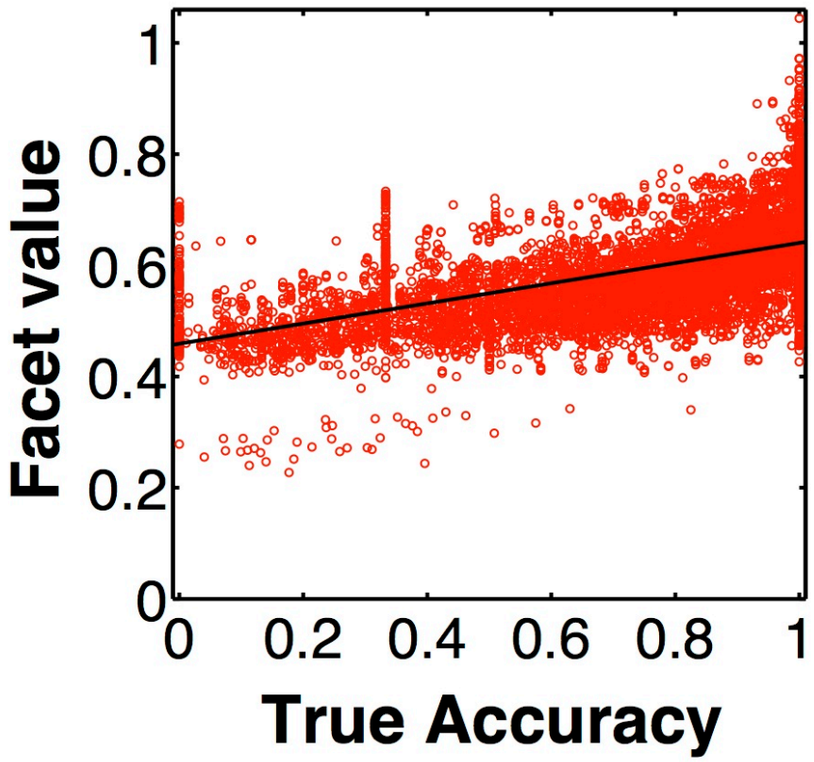
Experimental results

Best features trend well with accuracy.





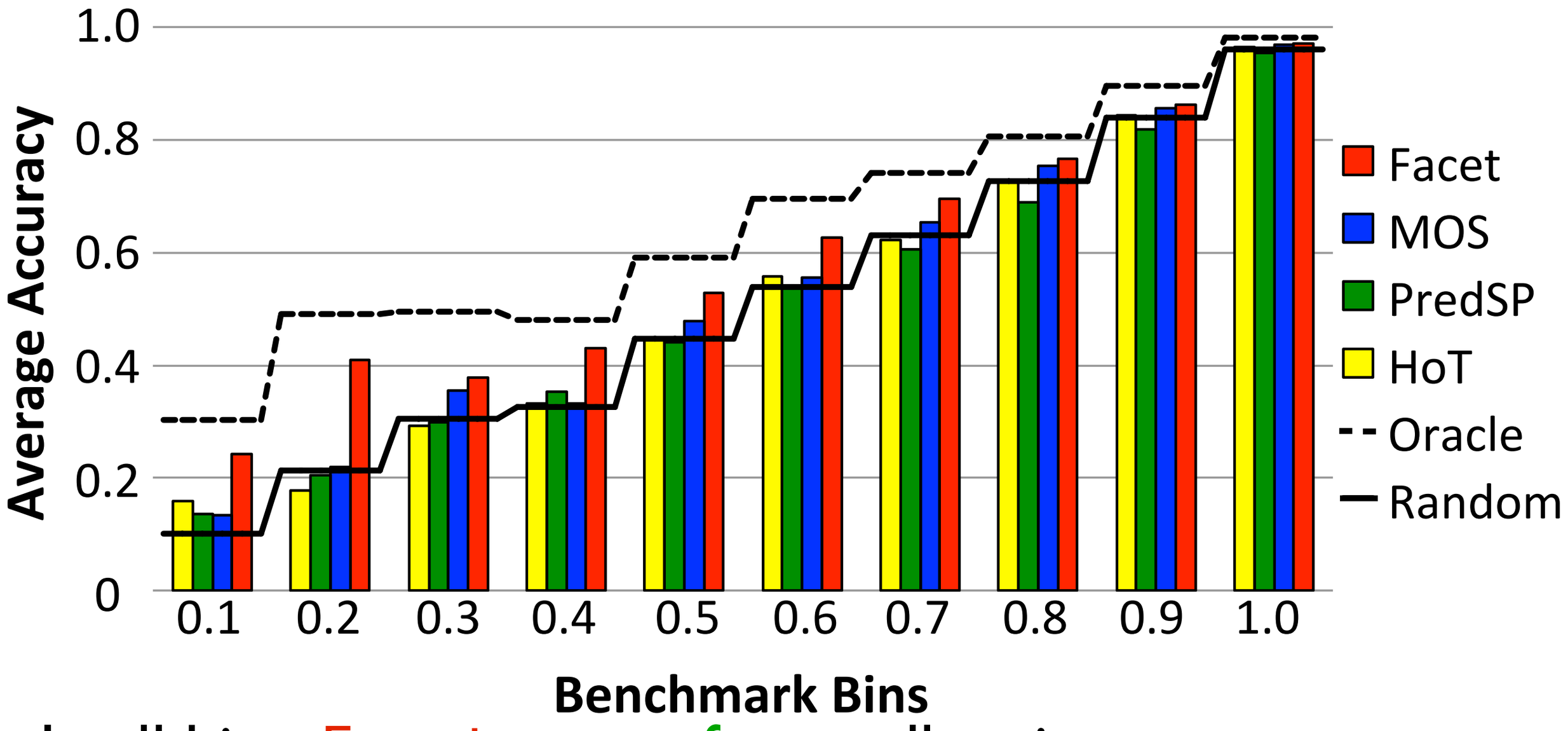




Facet estimator has better spread than its features.

Results

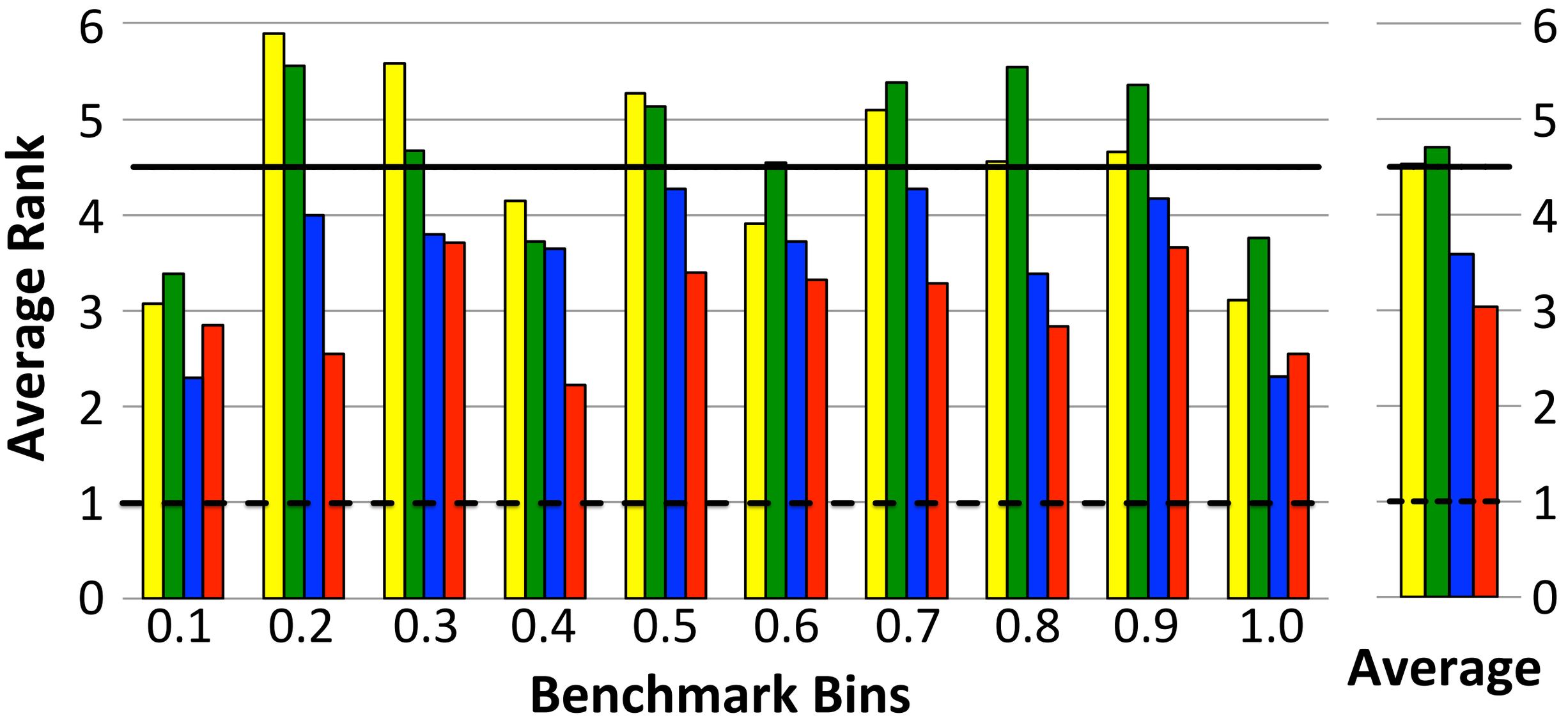
Average accuracy of advisors by default parameter bin



In all bins, Facet outperforms all estimators.

Results

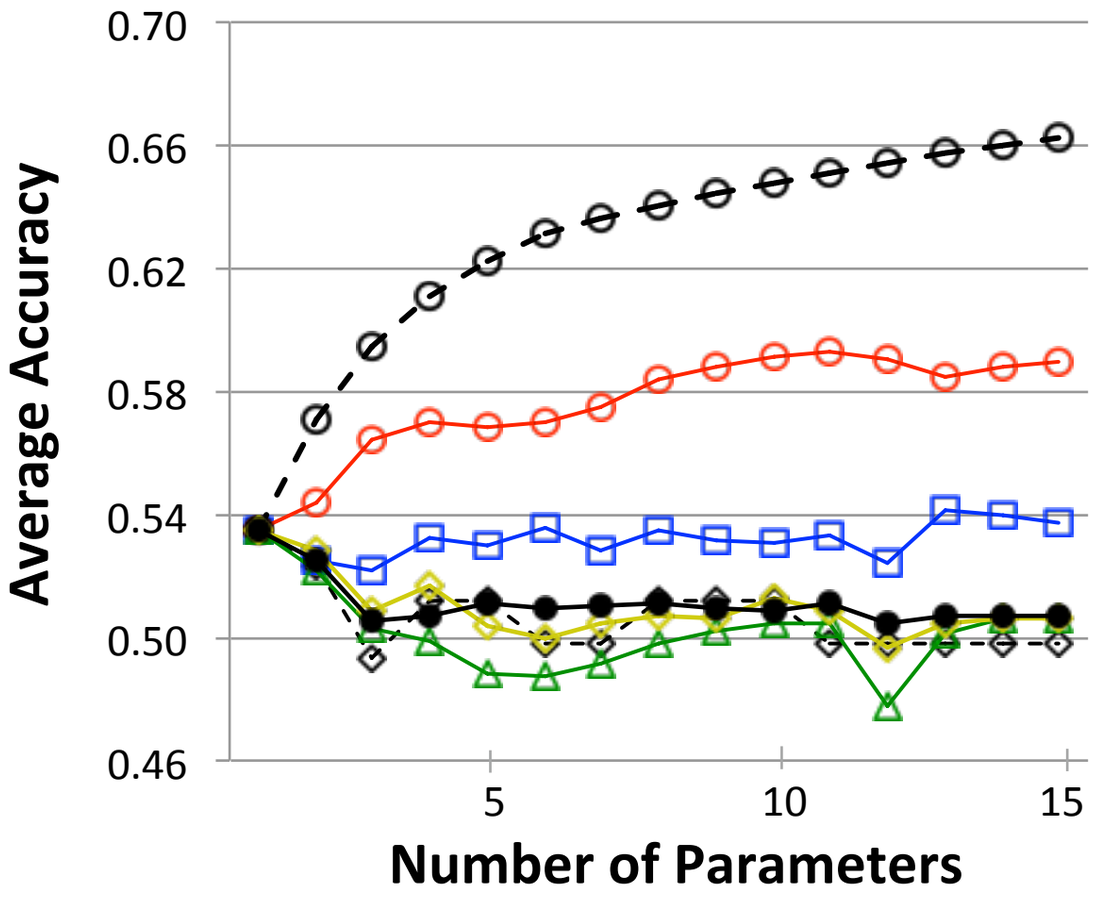
Average rank of advisors by default parameter bin

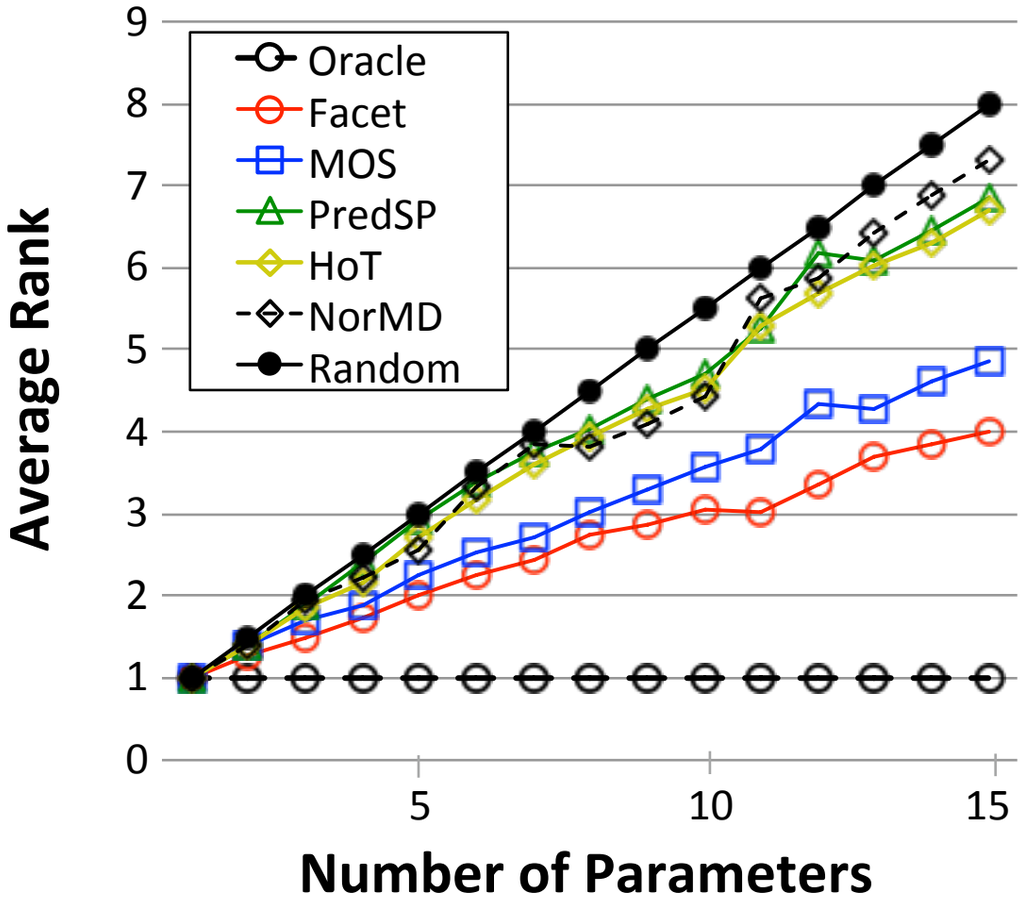


Facet has best rank, averaged across bins.

Results

Advisor performance versus parameter set cardinality





As the cardinality of P increases, Facet accuracy increases.

Conclusions

Facet yields a significant improvement for parameter advising.

- Estimator has best trend with true accuracy
- Parameter advisor gives 20% boost in accuracy over the default on hardest benchmarks
- Strictly better advising accuracy than other estimators across all bins
- Only estimator whose advisor benefits from more choices

Further research

- Develop a core column predictor for feature functions
- Find a stronger alignment gap feature
- Extend the estimator to DNA and RNA alignments
- Apply Facet to the problem of meta-alignment

Acknowledgments

People

Vladimir Filkov, UC Davis

Funding

- University of Arizona
  NSF IGERT in Genomics
  Grant DGE-0654435
- US NSF Grant IIS-1050293
- NSF Conference Travel Grant