FAST MOTIF DISCOVERY IN SHORT SEQUENCES

Honglei Liu, Fangqiu Han, Hongjun Zhou, Xifeng Yan, Kenneth S. Kosik

University of California, Santa Barbara

Motif discovery

- Motif: frequently appearing sequence patterns

Motif discovery

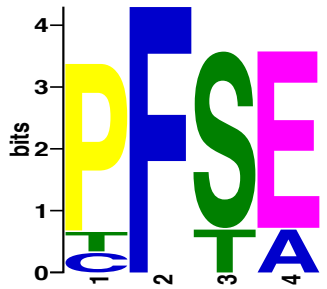
- Motif: frequently appearing sequence patterns
- Given a set of sequences S, the task of motif discovery is to identify sequence patterns that frequently appear in them

APFSELREIMHSYRG
PFSEEAYWHVGGMKA
LEWFESSGVPFSARS
RGIGSTLKPFSATRD
ATFSARWSNMVPDLR
CFSELPFSVWTPKAC
PFTEAGITADMWAWV

consensus string

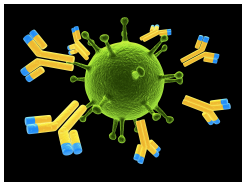
PFSE

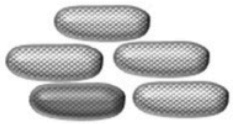
PWM (Position Weight Matrix)

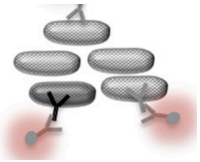


Applications

- Transcription factor binding sites (TFBSs) discovery
- Antibody biomarkers discovery



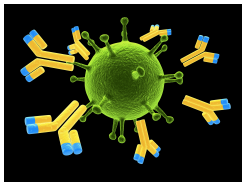


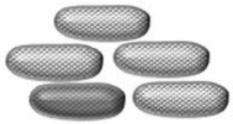


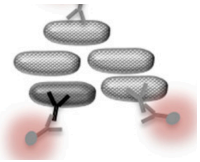
ESNTC**D**L**FVWQ**ACDGKQ
AEVACE**D**N**FVYQC**SDDW
SSASC**D**M**FVYQ**G**C**AEFN
RQGACV**D**D**YVYQC**GHFE
GHTACMTD**FV**H**QC**FPGT
PCV**DAFVYQ**QSGCNIA
RDGHCA**D**S**FVNQ**CVRPL
GRAACV**D**D**FVYQ**CVRQHE

Applications

- Transcription factor binding sites (TFBSs) discovery
- Antibody biomarkers discovery







ESNTCDLFVWQACDGKQ
AEVACEDNFVYQCSDDW
SSASCDMFVYQGCAEFN
RQGACVDDYVYQCGHFE
GHTACMTDFVHQCFPGT
PCVDAFVYQQSGCNIA
RDGHCADSFVNQCVRPL
GRAACVDDFVYQCVRQHE

Large scale, Large alphabet set, Short

New challenges

- Before next-generation sequencing era
  - At most several hundred sequences
- After next-generation sequencing era
  - Tens of thousands or even millions of sequences

New challenges

- Before next-generation sequencing era
  - At most several hundred sequences
- After next-generation sequencing era
  - Tens of thousands or even millions of sequences
- Existing methods fail to address the big data challenge (large scale, large alphabet set)

New challenges

- Before next-generation sequencing era
  - At most several hundred sequences
- After next-generation sequencing era
  - Tens of thousands or even millions of sequences
- Existing methods fail to address the big data challenge (large scale, large alphabet set)
  - MEME takes weeks to process 10k sequences

Framework design

- We have two options

Framework design

- We have two options
  - Design another motif finding algorithm

Framework design

- We have two options
  - Design another motif finding algorithm 
  - Reuse existing methods 

How to reuse existing methods

- Sampling method?

How to reuse existing methods

- Sampling method?
  - Low frequent motifs will be missed

How to reuse existing methods

- Sampling method?
  - Low frequent motifs will be missed
- Divide and conquer?

How to reuse existing methods

- Sampling method?
  - Low frequent motifs will be missed
- Divide and conquer?
  - Random partitioning does not work

How to reuse existing methods

- Sampling method?
  - Low frequent motifs will be missed
- Divide and conquer?
  - Random partitioning does not work
  - Global similarity does not work

RGIGSTLKPFSATRD

ATFSARWSNMVPDLR

How to reuse existing methods

- Sampling method?
  - Low frequent motifs will be missed
- Divide and conquer?
  - Random partitioning does not work
  - Global similarity does not work
  - Local similarity is needed

RGIGSTLKPFSATRD

ATFSARWSNMVPDLR

How to reuse existing methods

- Sampling method?
  - Low frequent motifs will be missed
- Divide and conquer?
  - Random partitioning does not work
  - Global similarity does not work
  - Local similarity is needed
  - Pairwise comparisons should be avoided

RGIGSTLKPFSATRD

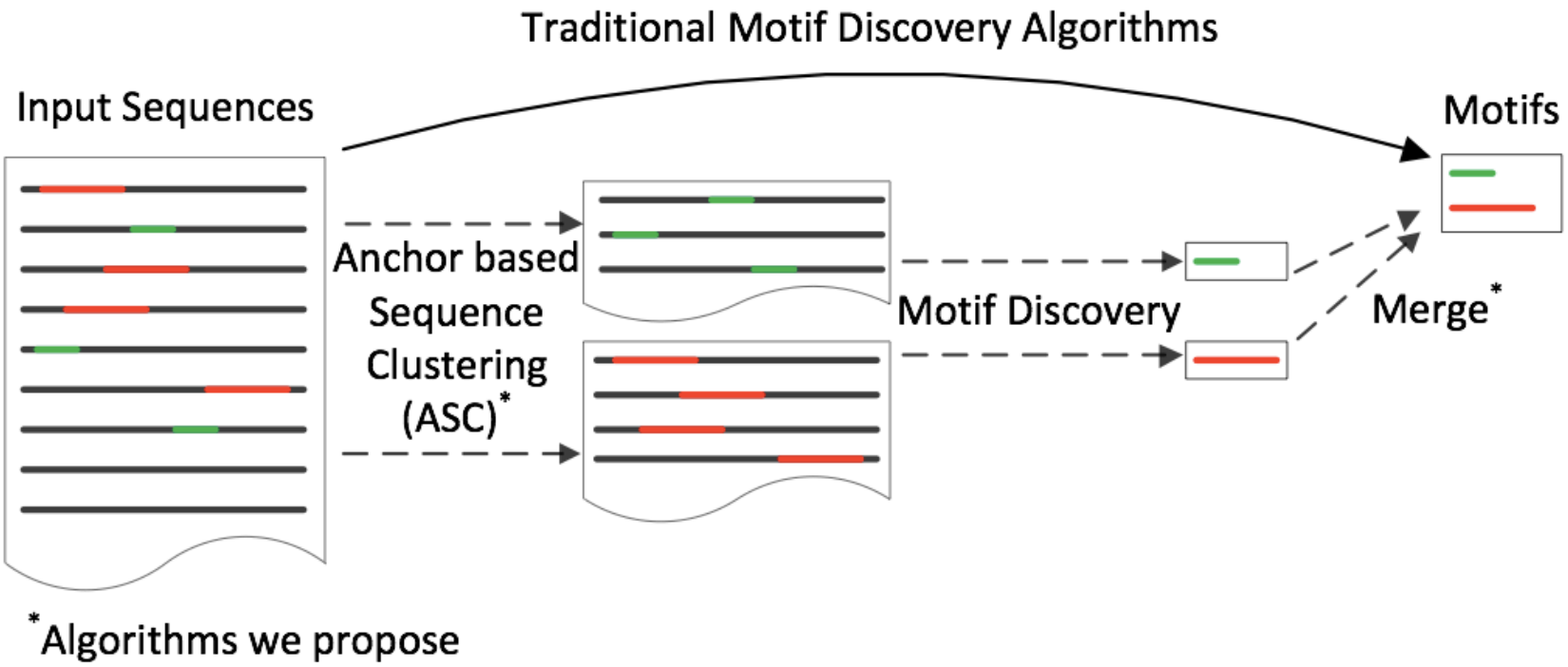
ATFSARWSNMVPDLR

How to reuse existing methods

- Straightforward methods do not work
  - Experiments with a real dataset of 11,642 sequences

Methods	# of motifs found	Runtime (Min.)
MEME	20	two weeks
Sampling	11	79
Partitioning	5	9
K-means	14	32

Our framework



Our clustering algorithm

- Anchor based Sequence Clustering algorithm (ASC)
  - Could capture local similarities
  - Avoid pairwise comparisons

Anchor based similarity

- Represent sequences as *q*-anchor sets
  - Gapped q-gram with variable shapes
  - e.g. 2-anchors of *PFSE* are {*PF, FS, SE, P_S, F_E, P_ _E*}

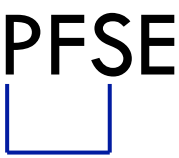
PFSE

Anchor based similarity

- Represent sequences as *q*-anchor sets
  - Gapped q-gram with variable shapes
  - e.g. 2-anchors of *PFSE* are {*PF, FS, SE, P_S, F_E, P_ _E*}

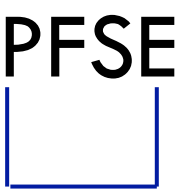
Anchor based similarity

- Represent sequences as *q*-anchor sets
  - Gapped q-gram with variable shapes
  - e.g. 2-anchors of *PFSE* are {*PF, FS, SE, P_S, F_E, P_ _E*}



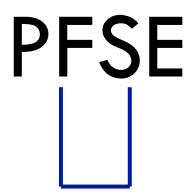
Anchor based similarity

- Represent sequences as *q*-anchor sets
  - Gapped q-gram with variable shapes
  - e.g. 2-anchors of *PFSE* are {*PF, FS, SE, P_S, F_E, P_ _E*}



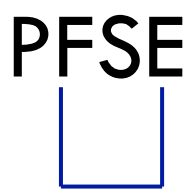
Anchor based similarity

- Represent sequences as *q*-anchor sets
  - Gapped q-gram with variable shapes
  - e.g. 2-anchors of *PFSE* are {*PF, FS, SE, P_S, F_E, P_ _E*}



Anchor based similarity

- Represent sequences as *q*-anchor sets
  - Gapped q-gram with variable shapes
  - e.g. 2-anchors of *PFSE* are {*PF, FS, SE, P_S, F_E, P_ _E*}

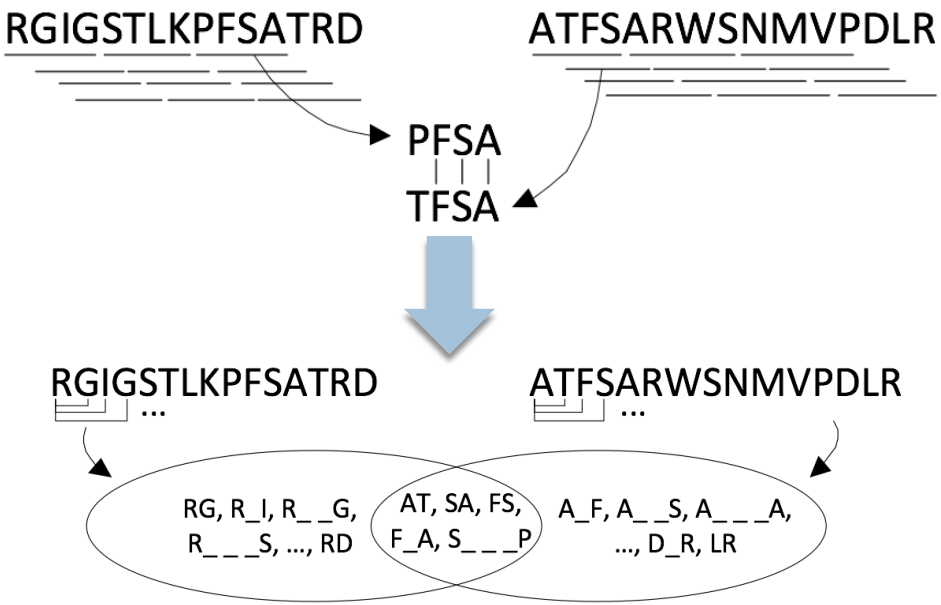


Anchor based similarity

- Represent sequences as *q*-anchor sets
  - Gapped q-gram with variable shapes
  - e.g. 2-anchors of *PFSE* are {*PF, FS, SE, P_S, F_E, P_ _E*}

Anchor based similarity

- Represent sequences as *q*-anchor sets
  - Gapped q-gram with variable shapes
  - e.g. 2-anchors of *PFSE* are {*PF, FS, SE, P_S, F_E, P_ _E*}
- Use anchor based similarity



Anchor based Sequence Clustering algorithm (ASC)

- Iterative process
  - Select cluster centers (*d* anchors)
  - Assign sequences to clusters

How to choose *d* anchors

- Theoretical analysis

How to choose *d* anchors

- Theoretical analysis
  - P_1: two sequence containing the same motif share *d* anchors

How to choose *d* anchors

- Theoretical analysis
  - P_1: two sequence containing the same motif share *d* anchors
  - P_2: two random sequences share *d* anchors

How to choose *d* anchors

- Theoretical analysis
  - P_1: two sequence containing the same motif share *d* anchors
  - P_2: two random sequences share *d* anchors
  - P_3: a random sequences contain *d* random anchors

How to choose *d* anchors

- Theoretical analysis
  - P_1: two sequence containing the same motif share *d* anchors
  - P_2: two random sequences share *d* anchors
  - P_3: a random sequences contain *d* random anchors
  - $P_1 >> P_2 >> P_3$

How to choose *d* anchors

- Theoretical analysis
  - P_1: two sequence containing the same motif share *d* anchors
  - P_2: two random sequences share *d* anchors
  - P_3: a random sequences contain *d* random anchors
  - $P_1 >> P_2 >> P_3$
- If we can choose *d* anchors that are from a motif, the clustering will be effective!

How to choose *d* anchors

- Significance of a motif
  - Over-representation is the key!

How to choose *d* anchors

- Significance of a motif
  - Over-representation is the key!
- Choose initial centers using *odd score*
  - Indicates how likely an anchor is from a motif

How to choose *d* anchors

- Significance of a motif
  - Over-representation is the key!
- Choose initial centers using *odd score*
  - Indicates how likely an anchor is from a motif
  - $P_{background}$: The probability of seeing an anchor by chance

How to choose *d* anchors

- Significance of a motif
  - Over-representation is the key!
- Choose initial centers using *odd score*
  - Indicates how likely an anchor is from a motif
  - $P_{background}$: The probability of seeing an anchor by chance
  - $P_{observed}$: The probability we observe

How to choose *d* anchors

- Significance of a motif
  - Over-representation is the key!
- Choose initial centers using *odd score*
  - Indicates how likely an anchor is from a motif
  - $P_{background}$: The probability of seeing an anchor by chance
  - $P_{observed}$: The probability we observe

odd score: $S(a) = \log P_{observed}(a) - \log P_{background}(a)$

$$P_{background}(a) = 1 - (1 - \prod_{\beta_i \in a} \theta_i)^{l-t+1}$$

$$P_{observed}(a) = \frac{f(a)}{N}$$

How to choose *d* anchors

- Adjust centers using *abundance score*

How to choose *d* anchors

- Adjust centers using *abundance score*
  - Indicates how unique an anchor is for a motif

How to choose *d* anchors

- Adjust centers using *abundance score*
  - Indicates how unique an anchor is for a motif
  - $P_{observe}$ within cluster: The observed probability of seeing an anchor in a cluster

How to choose *d* anchors

- Adjust centers using *abundance score*
  - Indicates how unique an anchor is for a motif
  - $P_{observe}$ within cluster: The observed probability of seeing an anchor in a cluster

Abundance score: $S_k(a) = \log \frac{f_k(a)}{N_k} - \log \frac{f(a)}{N}$

Experiments

- Five real datasets

Name	# of sequences	Length of sequences
Celiac	11,642	15
FXIIa	13,945	10
uPA	5,525	9
SrtA	4,993	8
PK	2,149	8

Experiments

- Five real datasets

Name	# of sequences	Length of sequences
Celiac	11,642	15
FXIIa	13,945	10
uPA	5,525	9
SrtA	4,993	8
PK	2,149	8

- Synthetic datasets
  - Plant motifs in sequences
  - Variable length, variable frequency, variable positions, *etc*

Experiments

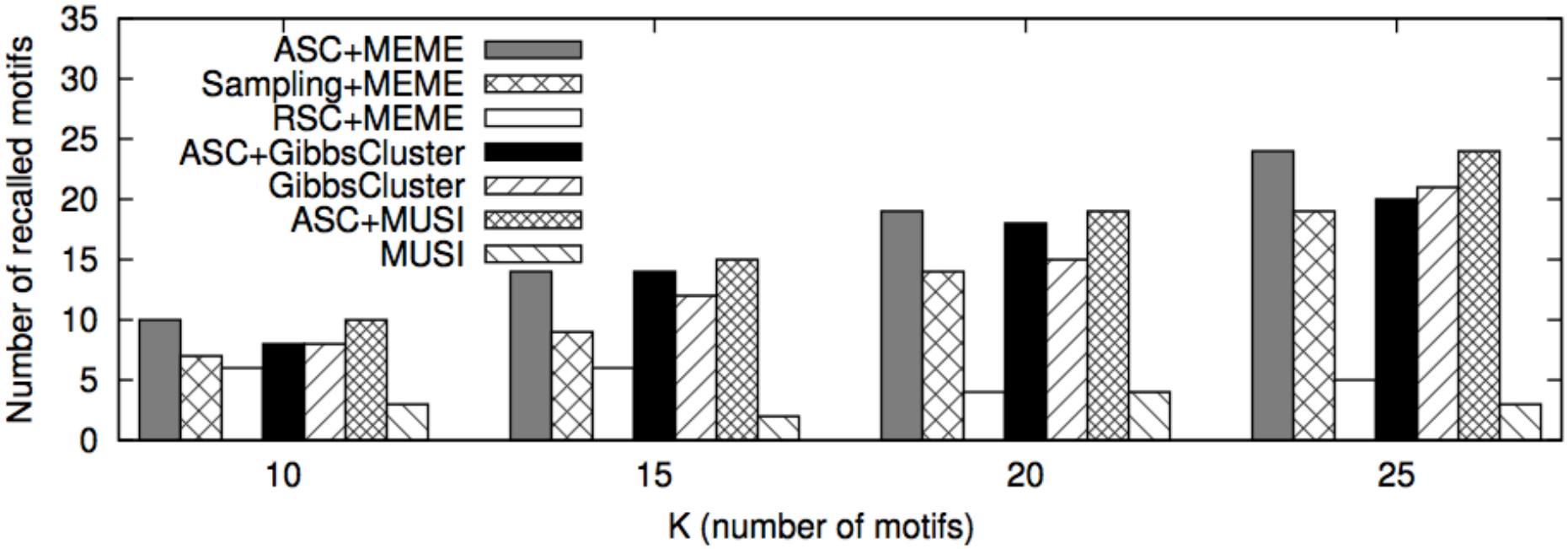
- Five real datasets

Name	# of sequences	Length of sequences
Celiac	11,642	15
FXIIa	13,945	10
uPA	5,525	9
SrtA	4,993	8
PK	2,149	8

- Synthetic datasets
  - Plant motifs in sequences
  - Variable length, variable frequency, variable positions, *etc*
- All the returned motifs are significant (precision=1)

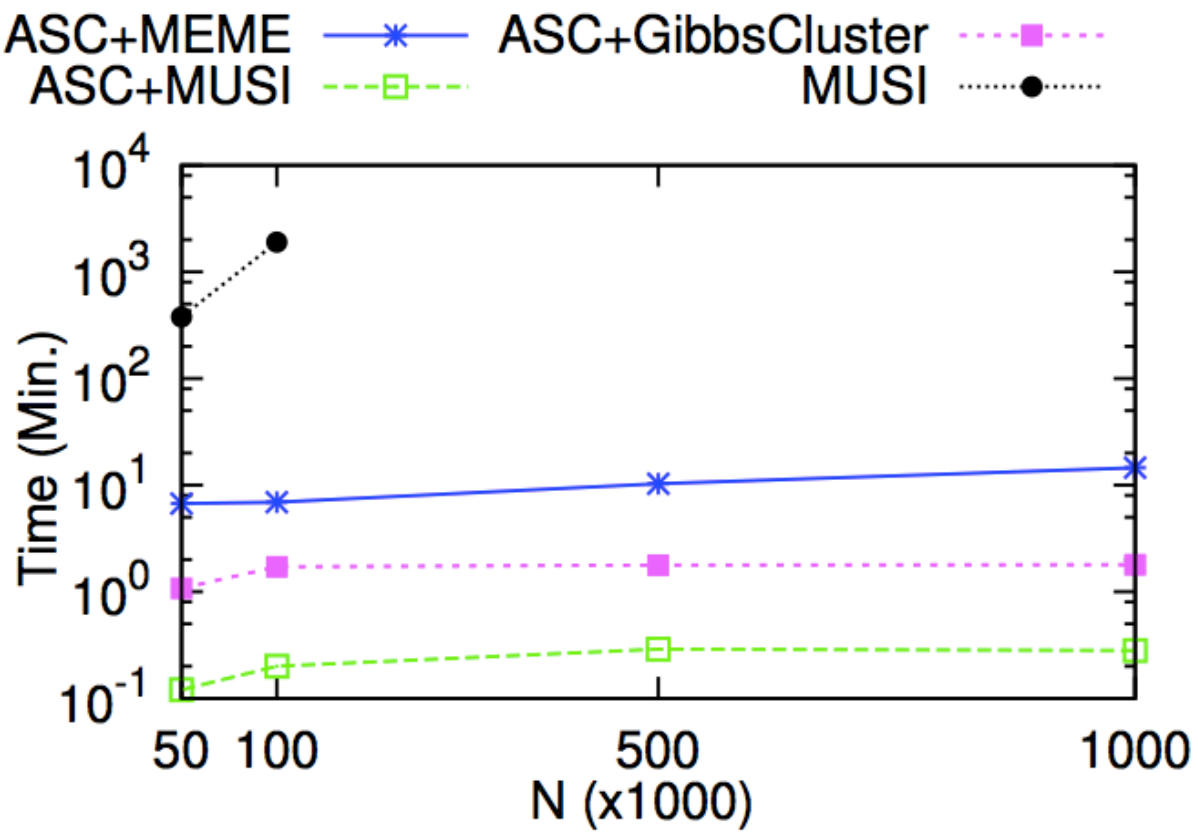
Number of recalled motifs

- Apply ASC on top of MEME, MUSI and GibbsCluster
- Number of recalled motifs from different methods using synthetic data (10k seq.)



Runtime

- Scalability



Real data

- Compare with MEME for *Celiac* dataset
  - 20 motifs were discovered by MEME
  - ASC-MEME could find even more motifs

# of clusters	# of motifs recalled	# of motifs found
10	17	16
20	18	19
40	20	22
60	20	24
w/o k	20	24

- MEME takes weeks
- ASC-MEME only takes minutes

Recap

- Big data challenge
- Reuse existing techniques
- Huge performance gain without losing accuracy

Thanks