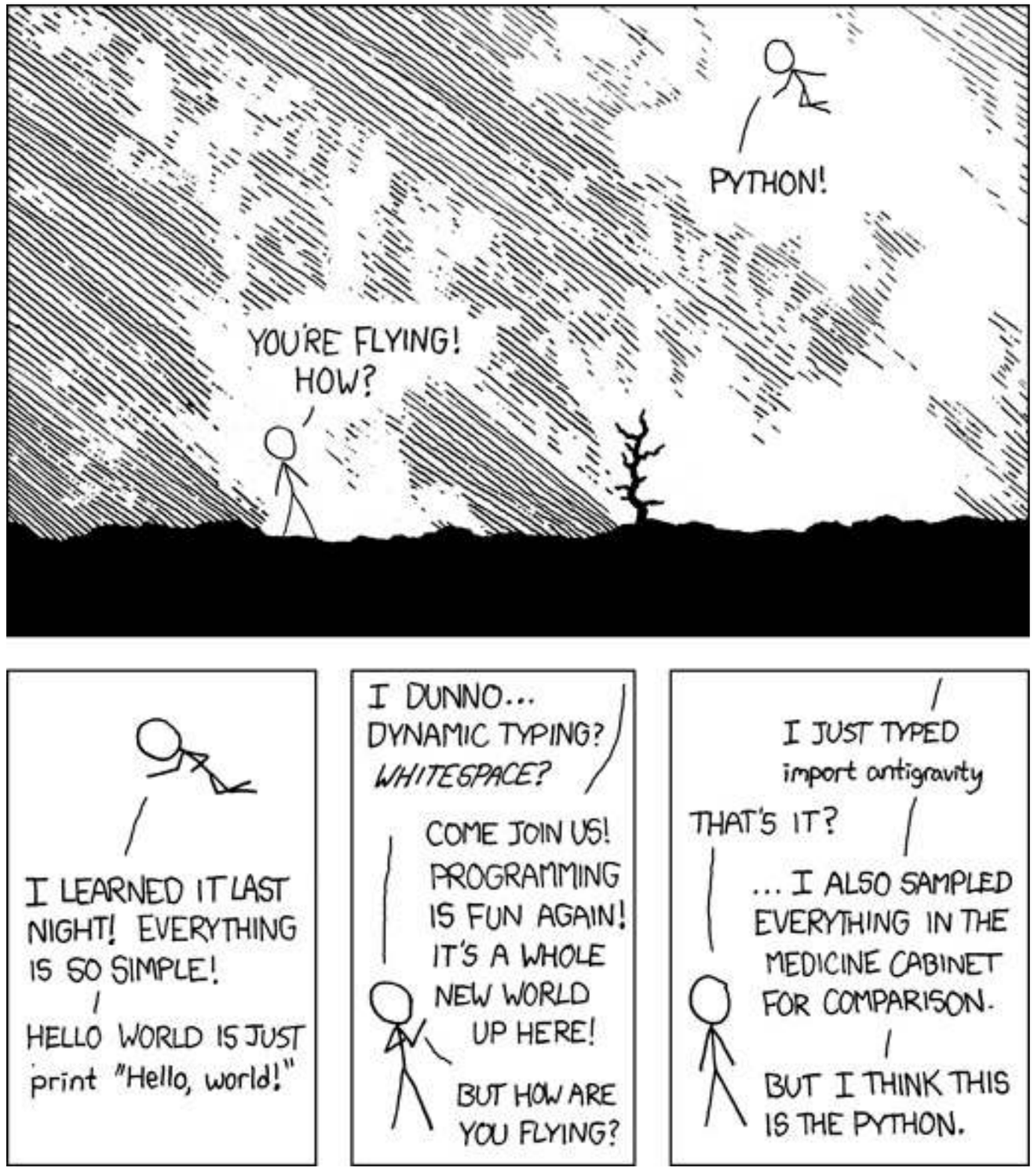
PyCogent

Greg Caporaso
gregcaporaso@gmail.com



http://xkcd.com/353/

PyCogent

- Support for interacting with external databases, many file formats, external applications
- Core objects for common data types (e.g., sequence, alignment, tree)
- Primary development takes place at CU Boulder and Australia National University

Why python?

- Object-oriented language with support for procedural programming
- Easily extensible with (e.g.) Cython to compile slow pieces
- Built-in support for testing via the unittest and doctest modules

PrimerProspector

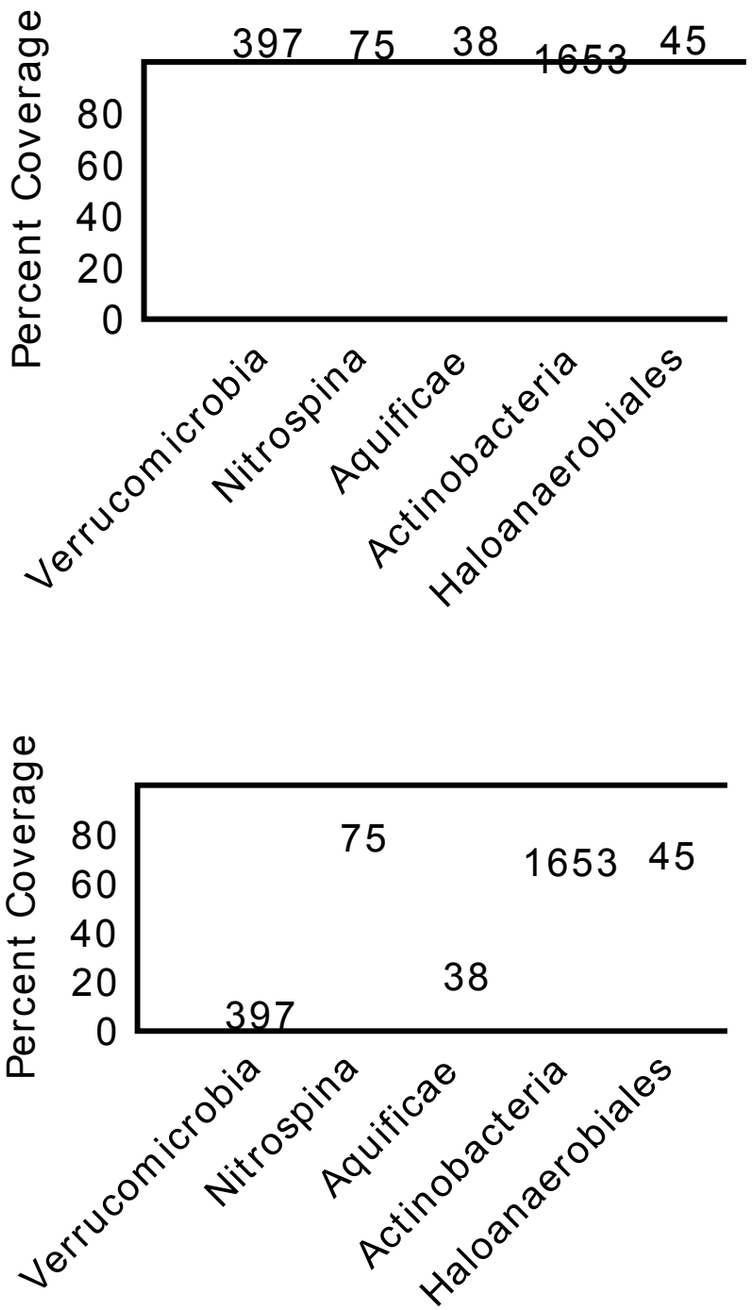
analysis and de novo design of primers
for barcoded amplicon sequencing

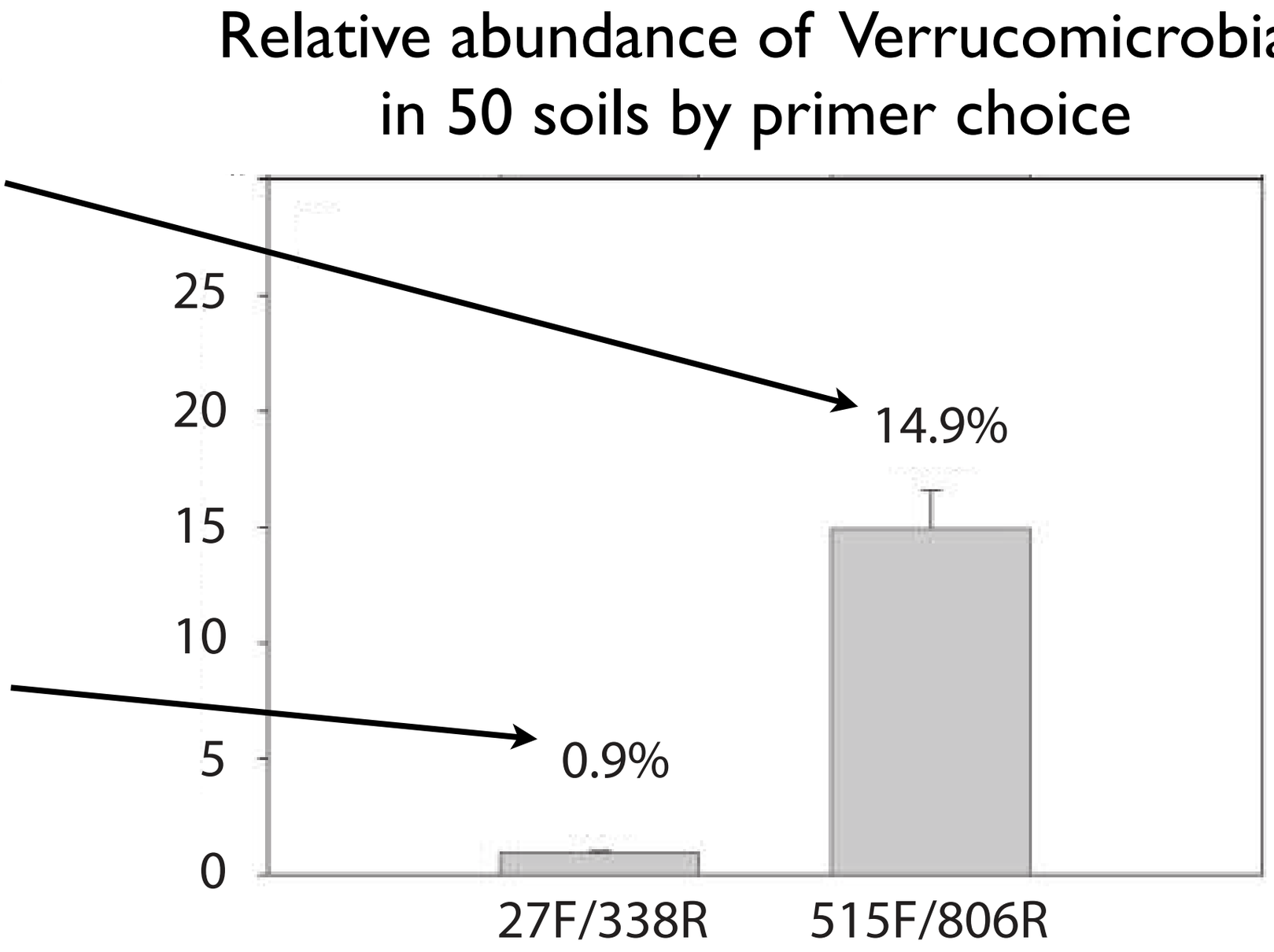
http://pprospector.sourceforge.net/

- evaluate primers for coverage
- evaluate amplicons for informativeness
- test barcodes for secondary structure or primer/dimer formation

PrimerProspector: de novo design and taxonomic analysis of PCR primers
WA Walters, JG Caporaso, CL Lauber, D Berg-Lyons, N Fierer, R Knight
Bioinformatics (2011) 27(8): 1159-1161

PrimerProspector and the Verrucomicrobia





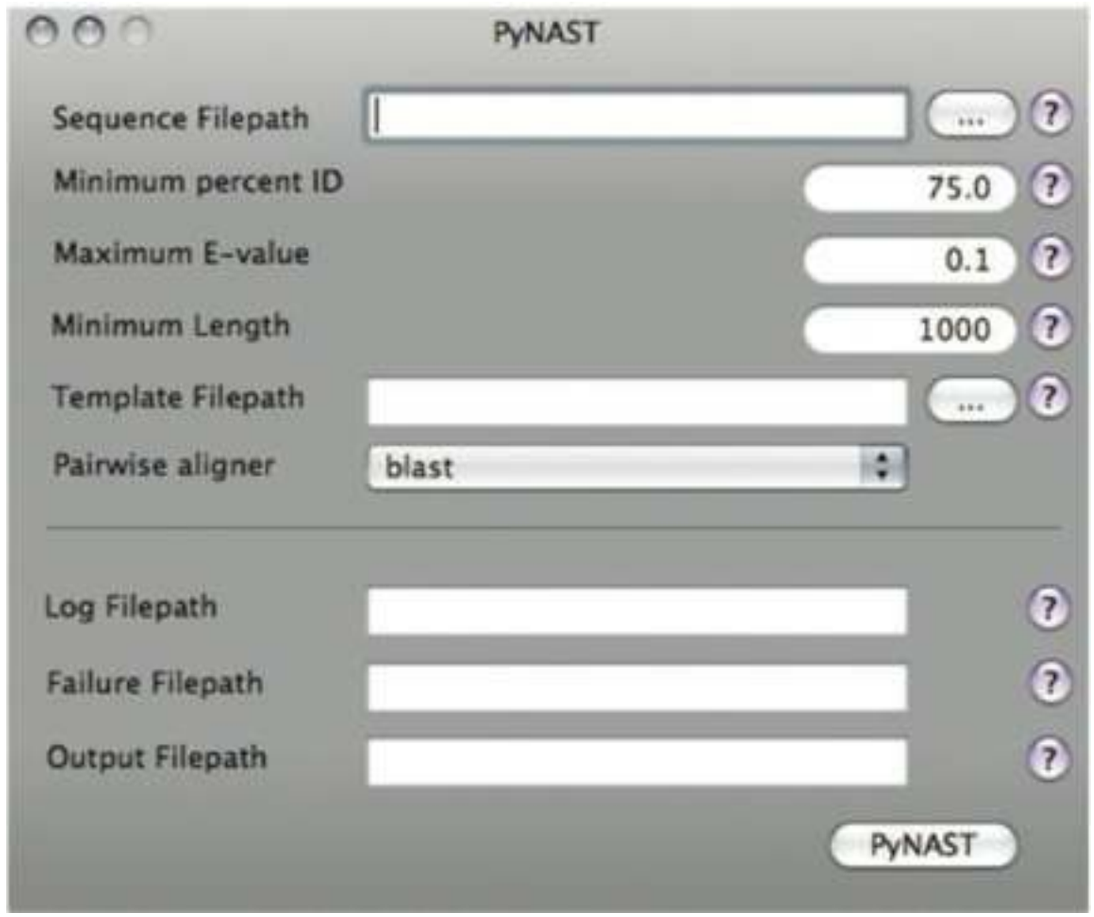
GT Bergmann, ST Bates, KG Eilers, CL Lauber, JG Caporaso, WA Walters, R Knight, N Fierer
The under-recognized dominance of Verrucomicrobia in soil bacterial communities
Soil Biology and Biochemistry, In Press.

PyNAST

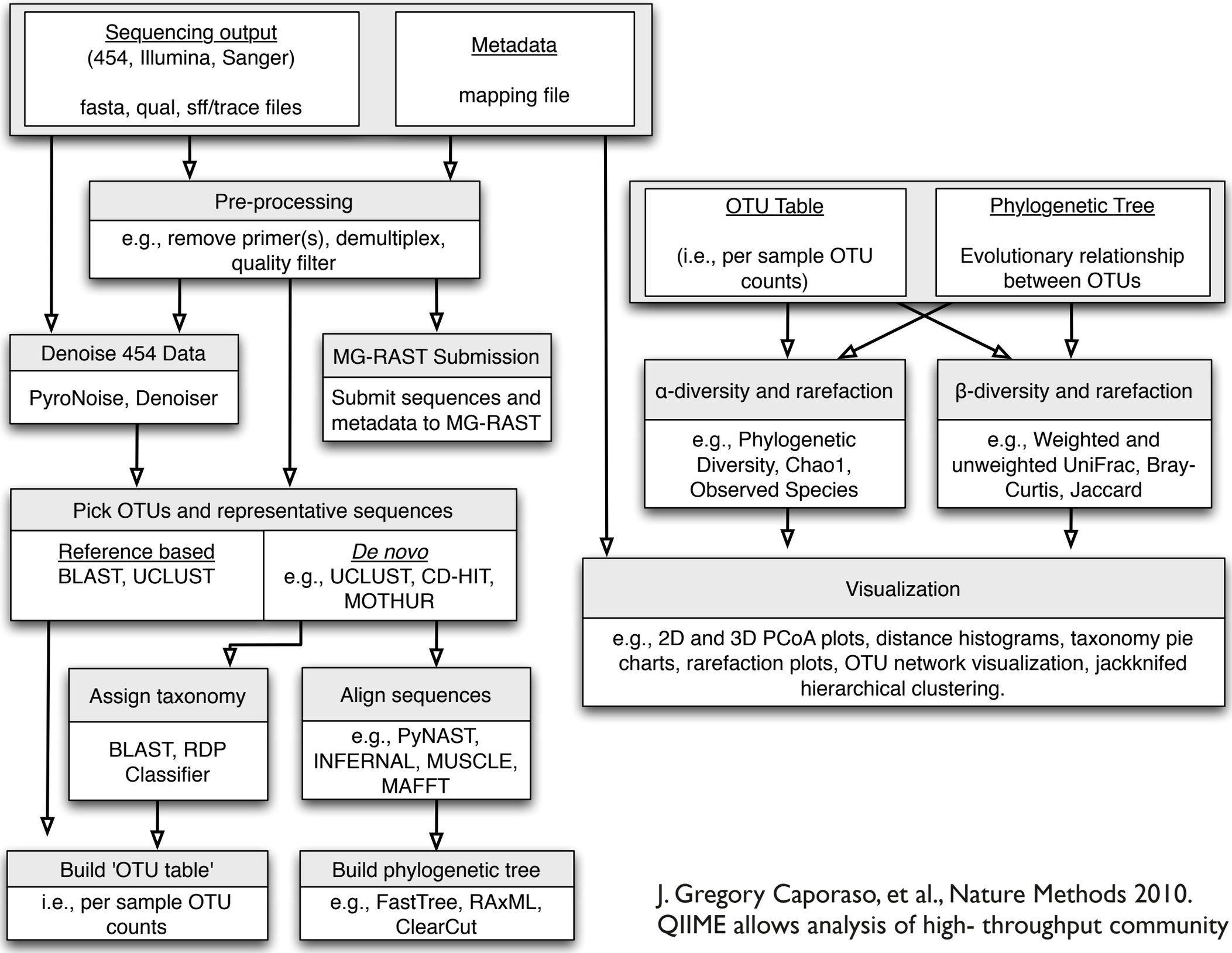
Python Nearest Alignment Space Termination

http://pynast.sourceforge.net

tool for aligning sequences to a template alignment



Quantitative Insights Into Microbial Ecology



J. Gregory Caporaso, et al., Nature Methods 2010.
QIIME allows analysis of high- throughput community sequencing data

Lecture outline

- Features of PyCogent: building a tree of life
- Usage example: Applying PyCogent to study sequence coevolution.
- Defining a PyCogent script.
- Building a new ApplicationController object: formatdb.

PyCogent usage: building a tree of life

http://bit.ly/pQ0FSp

See Woese 1987 *(Bacterial Evolution)* and Woese 1990 *(Towards a natural system of organisms)* for discussion of the ideas presented in this example.

PyCogent usage: building a tree of life

1. Obtain a collection of archaeal, bacterial, and eukaryotic 16S sequences from NCBI.
2. Perform a multiple sequence alignment on the sequences.
3. Build a tree from the sequences.
4. Confirm that there are three distinct clusters in the tree, supporting the idea that cellular life on Earth clusters into three domains detectable by distances between 16S sequences.
5. Output a graphic of the tree as a PDF.

Summary: building a tree of life

Compile sequences:

EUtils module

Load sequences:

LoadSeqs, SequenceCollection object, DNA

Align sequences:

The Muscle ApplicationController object

Build a tree:

The FastTree ApplicationController

The PhyloNode object

Visualize the tree:

UnrootedDenrogram object, and the cogent drawing functionality

The PyCogent Application Controller Framework

- Run external application available via PyCogent, and therefore can integrate external tools into your bioinformatics pipeline
  - Define common interface for applications
  - Example includes steps we ran through in the tree of life application: muscle and fast tree.

Full documentation at:
http://pycogent.sourceforge.net/examples/application_controller_framework.html

Common interface for third-party multiple sequence aligners

```
caporaso@mentat app> egrep 'def align_unaligned_seqs' *py

clearcut.py:def align_unaligned_seqs(seqs, moltype, params=None):
clustalw.py:def align_unaligned_seqs(seqs, moltype, params=None):
mafft.py:def align_unaligned_seqs(seqs,moltype,params=None,accurate=False):
muscle.py:def align_unaligned_seqs(seqs, moltype, params=None):
```

```
caporaso@mentat app> egrep 'def build_tree_from_alignment' *py

clearcut.py:def build_tree_from_alignment(aln, moltype, best_tree=False, params={},\
clustalw.py:def build_tree_from_alignment(aln, moltype, best_tree=False, params=None):
fasttree.py:def build_tree_from_alignment(aln, moltype, best_tree=False, params=None):
fasttree_v1.py:def build_tree_from_alignment(aln, moltype, best_tree=False, params=None):
mafft.py:def build_tree_from_alignment(aln, moltype, best_tree=False, params={},\
muscle.py:def build_tree_from_alignment(aln, moltype, best_tree=False, params=None):
raxml.py:def build_tree_from_alignment(aln, moltype, best_tree=False, params={}):
```

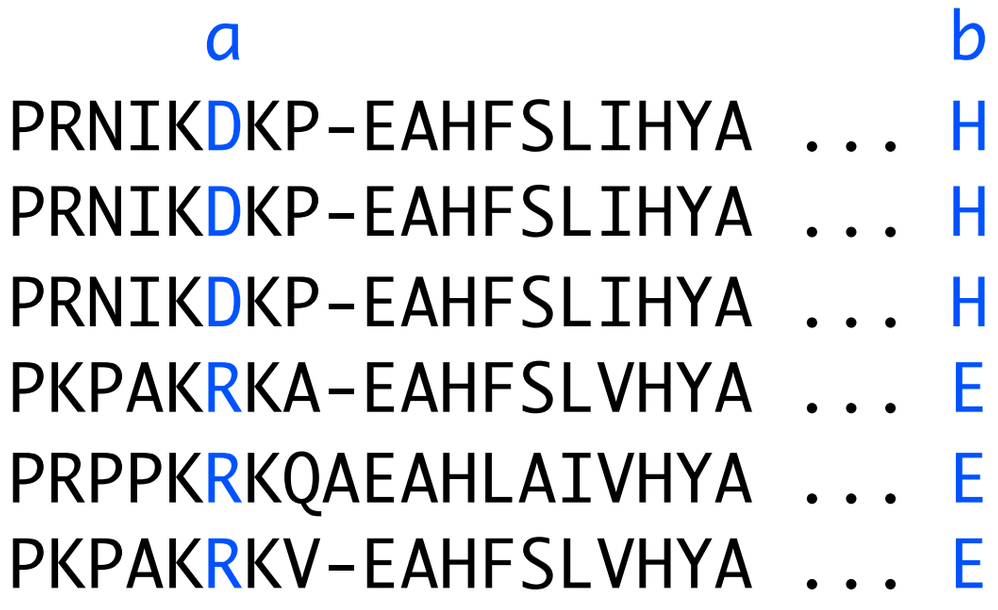
How to control third party applications with PyCogent

- Check the applications which are currently supported. (If yours isn't new application controllers are typically easy to write.)
- The application must be installed and in your search path.

```
caporaso@saguaro app> pwd
/Users/caporaso/code/PyCogent_svn/cogent/app
caporaso@saguaro app> ls
__init__.py             dotur.py            mothur.py           rnashapes.py
blast.py                dynalign.py         msms.py             rnaview.py
carnac.py               fasttree.py         muscle.py           sfffile.py
cd_hit.py               fasttree_v1.py      nupack.py           sffinfo.py
clearcut.py             foldalign.py        parameters.py       sfold.py
clustalw.py             formatdb.py         pfold.py            stride.py
cmfinder.py             gctmpca.py          pknotsrg.py         uclust.py
comrna.py               ilm.py              raxml.py            unafold.py
consan.py               infernal.py         rdp_classifier.py   util.py
contrafold.py           knetfold.py         rdp_classifier20.py vienna_package.py
cove.py                 mafft.py            rnaalifold.py
dialign.py              mfold.py            rnaforester.py
```

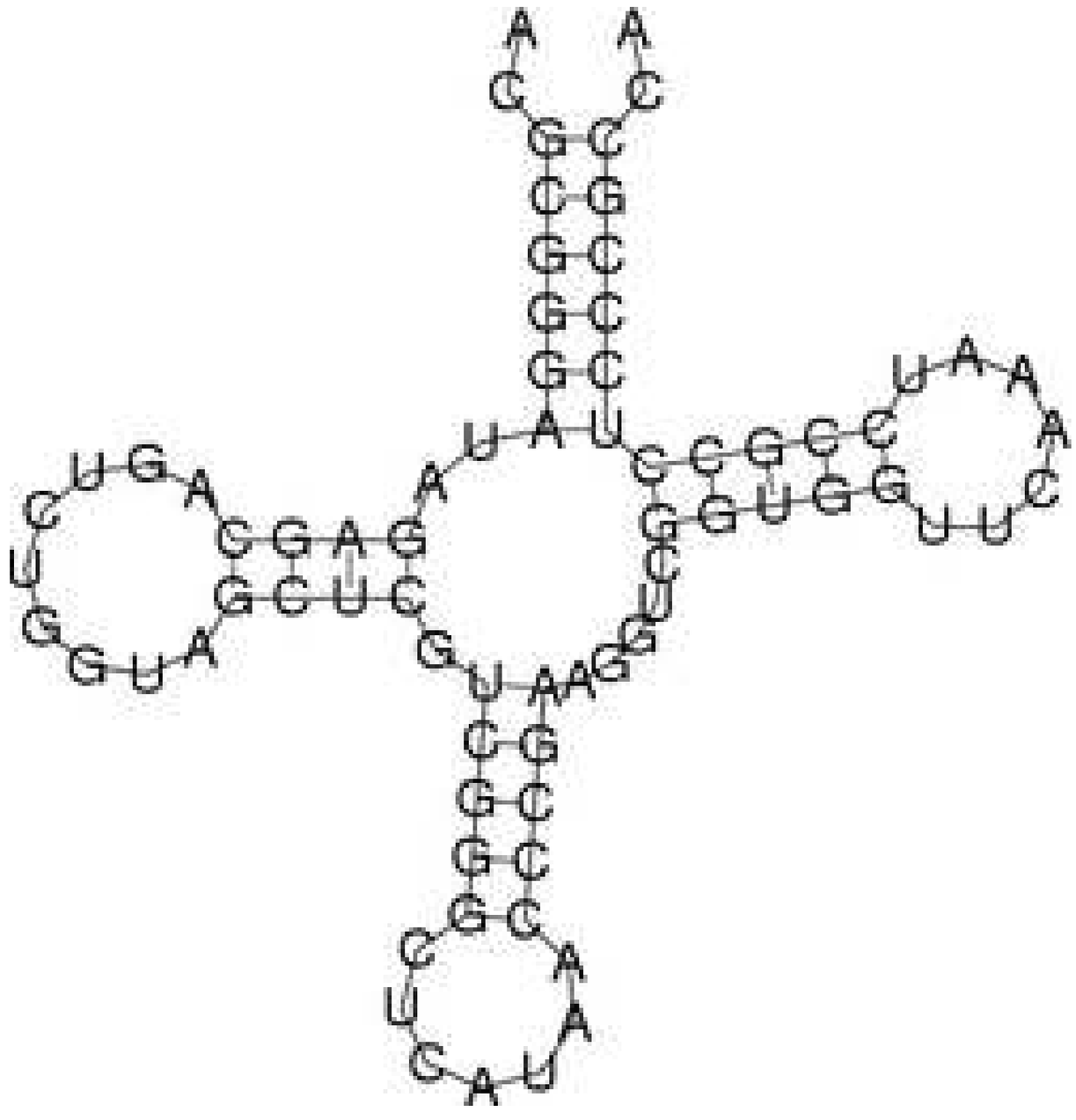
Case study: sequence coevolution

```
      a                          b
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PKPAKRKA-EAHFSLVHYA ... E
PRPPKRKQAEAHLAIVHYA ... E
PKPAKRKV-EAHFSLVHYA ... E
```



covariation refers to the correlated pattern;
coevolution refers to a possible cause of the pattern

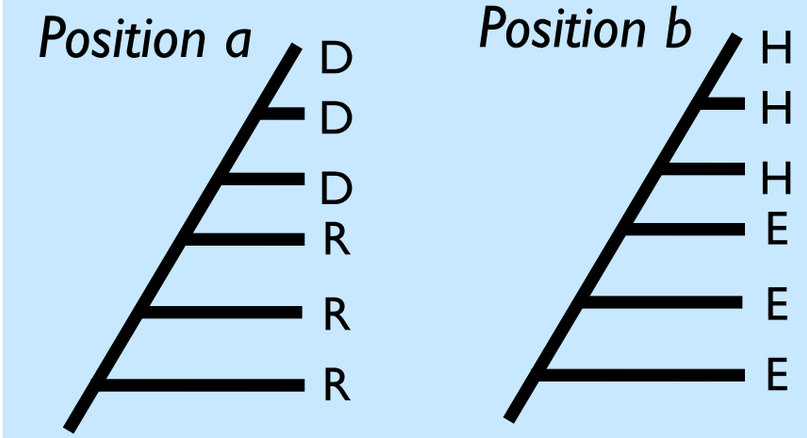
Base pairs in ncRNAs



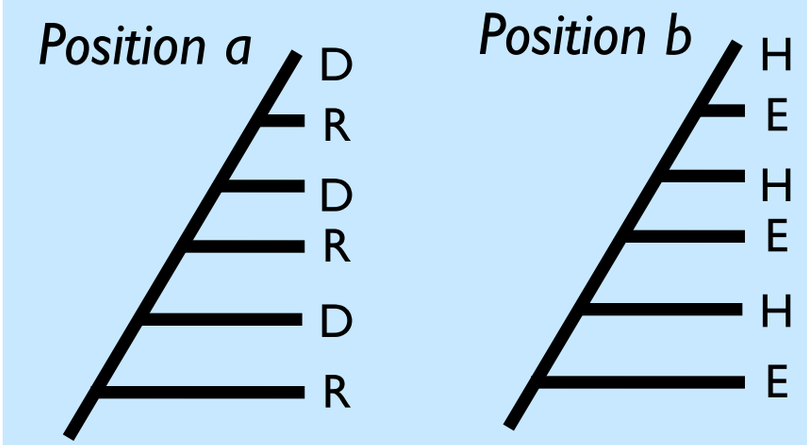
tRNA$_{Met}$ secondary structure

```
     a                          b
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PKPAKRKA-EAHFSLVHYA ... E
PRPPKRKQAEAHLAIVHYA ... E
PKPAKRKV-EAHFSLVHYA ... E
```

Less evidence for functional significance



More evidence for functional significance



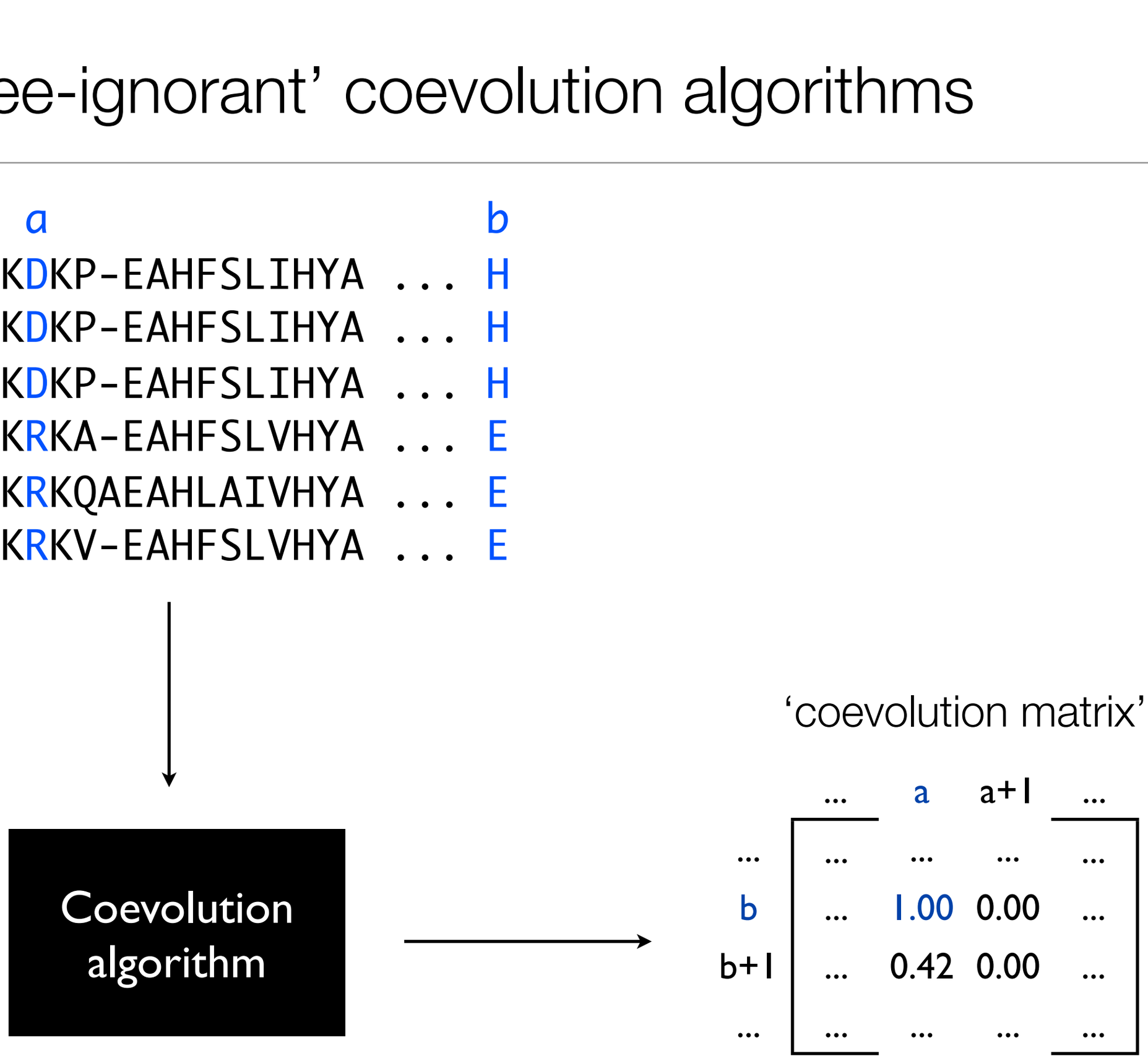
Coevolution algorithm

‘coevolution matrix’

	...	a	a+1	...
...	...	...	...	...
b	...	1.00	0.00	...
b+1	...	0.42	0.00	...
...	...	...	...	...

‘Tree-ignorant’ coevolution algorithms

```
     a                      b
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PKPAKRKA-EAHFSLVHYA ... E
PRPPKRKQAEAHLAIVHYA ... E
PKPAKRKV-EAHFSLVHYA ... E
```



Normalized Mutual Information

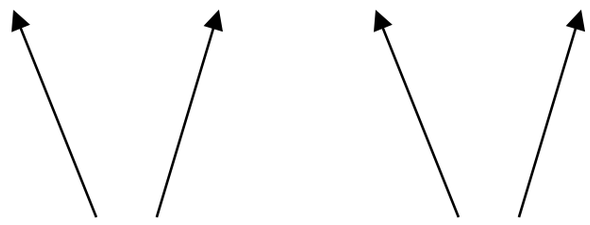
$$H(X) = -\sum_{x} p_x \log_2(p_x)$$

$$H(X,Y) = -\sum_{x,y} p_{x,y} \log_2(p_{x,y})$$

$$MI(X;Y) = H(X) + H(Y) - H(X,Y)$$

$$NMI(X;Y) = \frac{MI(X;Y)}{H(X,Y)}$$

A	A	A	C
A	C	A	C
A	D	A	C
A	E	A	C
A	F	F	D
A	G	F	G
C	H	F	G
Entropy:	Low	High	Med.	Med.

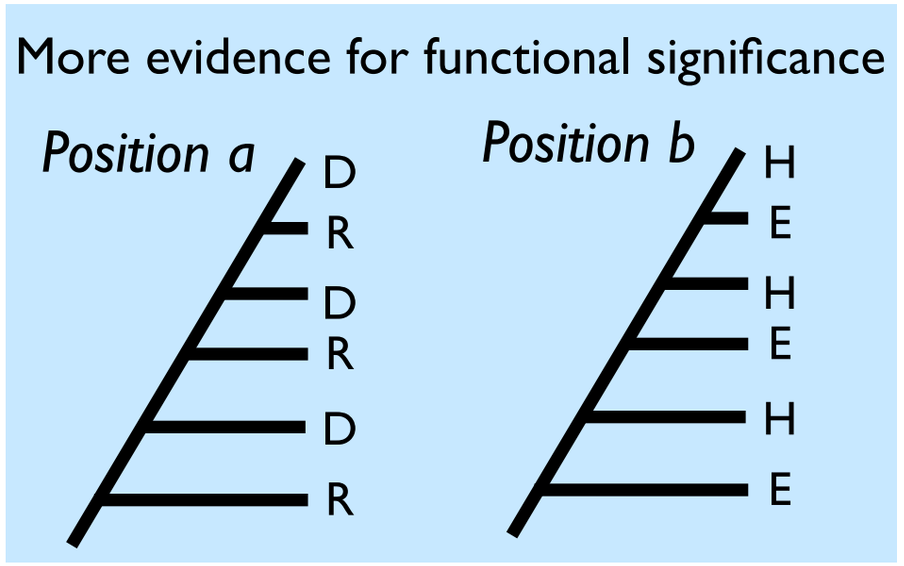


Lower MI Higher MI

Martin, L.C.; Gloor, G.B.; Dunn, S.D. & Wahl, L.M. (2005), 'Using information theory to search for co-evolving residues in proteins.', *Bioinformatics* **21**(22), 4116--4124.

‘Tree-aware’ coevolution algorithms

```
     a                          b
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PKPAKRKA-EAHFSLVHYA ... E
PRPPKRKQAEAHLAIVHYA ... E
PKPAKRKV-EAHFSLVHYA ... E
```

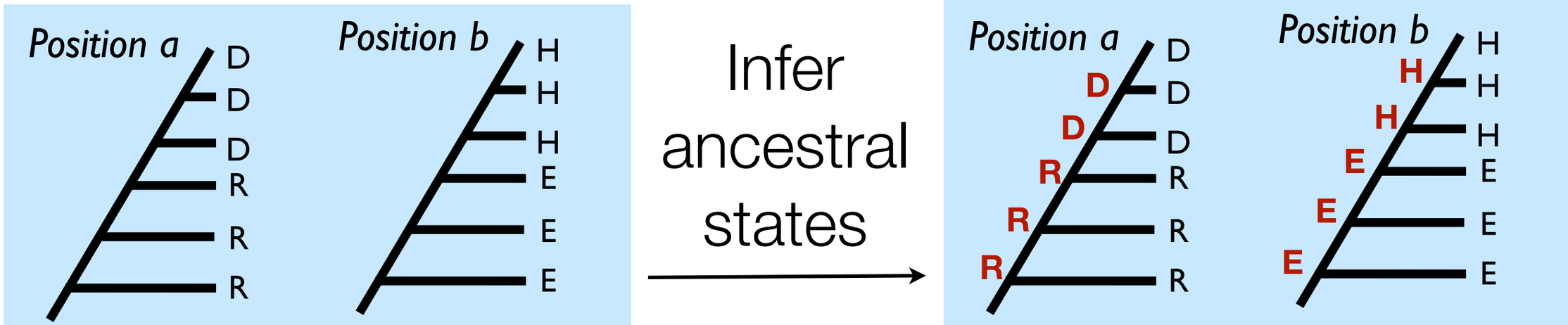


Coevolution algorithm

‘coevolution matrix’

	...	a	a+1	...
...	...	...	...	...
b	...	1.00	0.00	...
b+1	...	0.42	0.00	...
...	...	...	...	...

Ancestral States



Count correlated changes, giving more weight to changes occurring closer in evolutionary time.

```
score = 0
for each pair of sequences:
    if residue at both positions has changed since LCA:
      score = score +
       (branch length between sequences)^-1
```

Tree-aware versus tree-ignorant algorithms

Tree-aware		**Tree-ignorant**
slow		fast (relatively)
explicit control for effect of ancestry should result in fewer spurious hits		confound effect of ancestry with coevolutionary signal

Reduced-state alphabets: recode 20 amino acids in fewer states based on a selected physicochemical property

```
     a                    b
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PKPAKRKA-EAHFSLVHYA ... E
PRPPKRKQAEAHLAIVHYA ... E
PKPAKRKV-EAHFSLVHYA ... E
```

3-state charge alphabet →

```
          a                          b
UPUUPNPU-NUPUUUUPUU ... P
UPUUPNPU-NUPUUUUPUU ... P
UPUUPPPU-NUPUUUUPUU ... N
UPUUPPPUUNUPUUUUPUU ... N
UPUUPPPU-NUPUUUUPUU ... N
```

‘CHARGE_HIS_3’ alphabet:

P(ositive) ← RHK

N(egative) ← DE

U(ncharged) ← ACFGJLMNPQSTVWY

Improve statistical power by hiding variation among similar residues

```
     a                    b
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PKPAKRKA-EAHFSLVHYA ... E
PRPPKRKQAEAHLAIVHYA ... E
PKPAKRKV-EAHFSLVHYA ... E
```

3-state charge alphabet

```
     a                    b
PRNIKDKP-EAHFSLIHYA ... R
PRNIKEKP-EAHFSLIHYA ... K
PRNIKEKP-EAHFSLIHYA ... H
PKPAKRKA-EAHFSLVHYA ... E
PRPPKHKQAEAHLAIVHYA ... D
PKPAKKKV-EAHFSLVHYA ... E
```

```
          a                          b
UPUUPNPU-NUPUUUUPUU ... P
UPUUPNPU-NUPUUUUPUU ... P
UPUUPPPU-NUPUUUUPUU ... N
UPUUPPPUUNUPUUUUPUU ... N
UPUUPPPU-NUPUUUUPUU ... N
```

Improve statistical power by hiding variation among similar residues

```
     a          !          b
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PRNIKDKP-EAHFSLIHYA ... H
PKPAKRKA-EAHFSLVHYA ... E
PRPPKRKQAEAHLAIVHYA ... E
PKPAKRKV-EAHFSLVHYA ... E
```

3-state charge alphabet

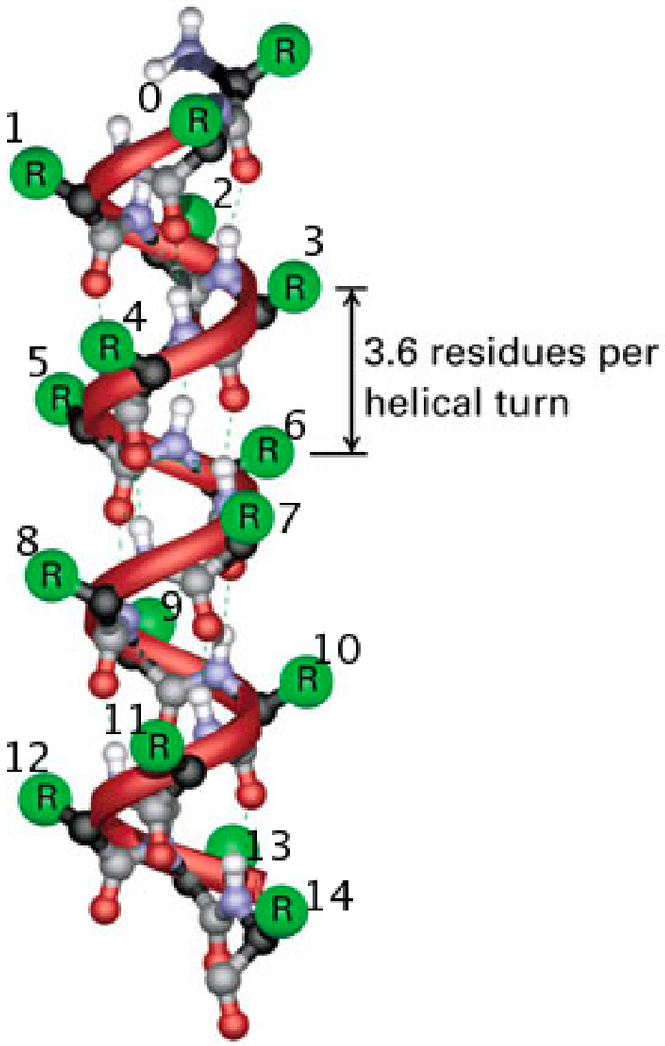
```
     a          ✓          b
UPUUPNPU-NUPUUUUPUU ... P
UPUUPNPU-NUPUUUUPUU ... P
UPUUPPPU-NUPUUUUPUU ... N
UPUUPPPUUNUPUUUUPUU ... N
UPUUPPPU-NUPUUUUPUU ... N
```

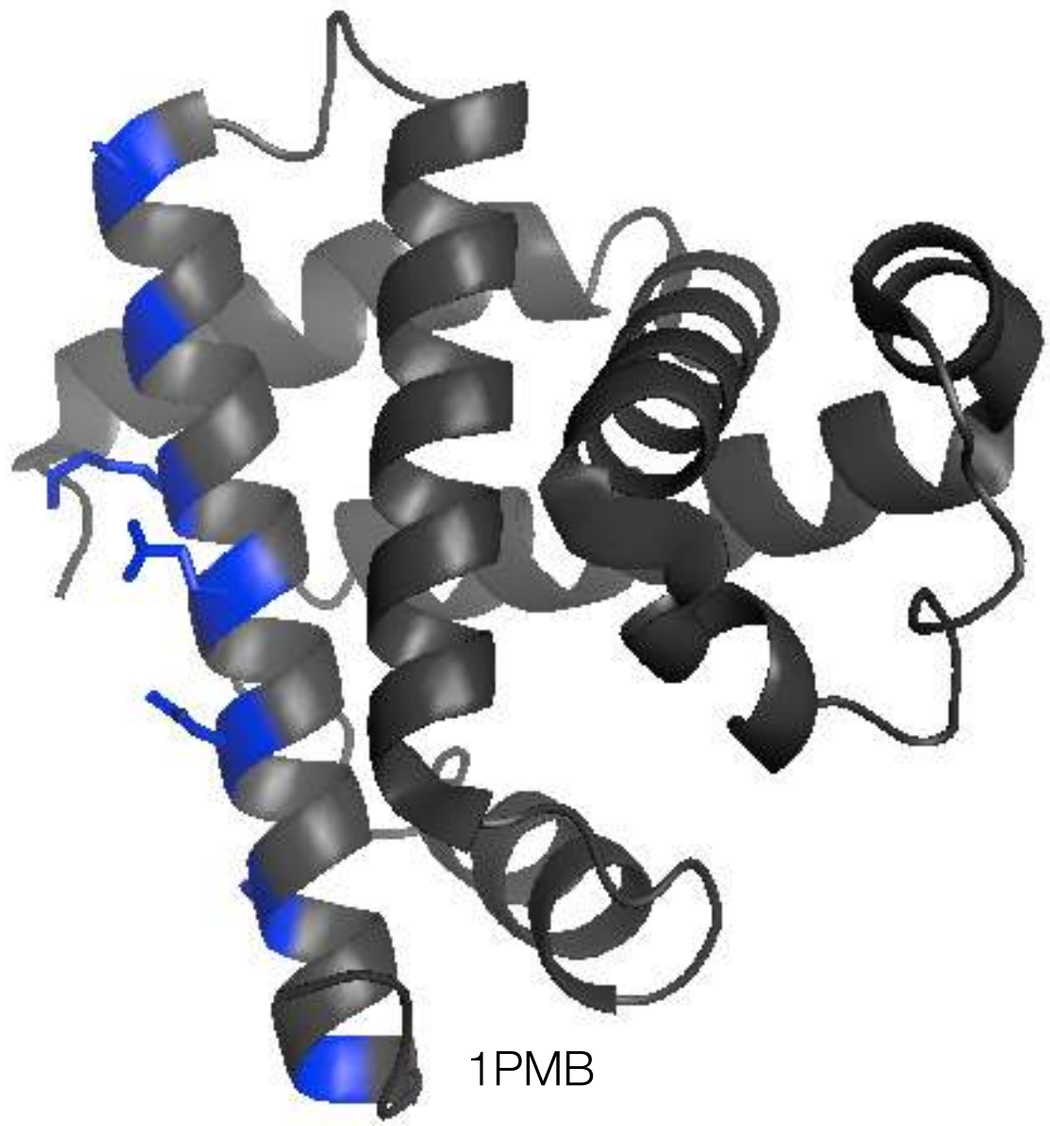
Detecting Coevolution Without Phylogenetic Trees? Tree-Ignorant Metrics Of Coevolution Perform As Well As Tree-Aware Metrics.

Comparing tree-ignorant coevolution scores against background distribution of scores with same embedded ancestry could circumvent the need for explicit incorporation of a tree.

Recoding alignments in alphabets modeling appropriate physicochemical properties should improve statistical power of coevolution algorithms.

Novel comparison on protein secondary structure: stacked residues in alpha helices





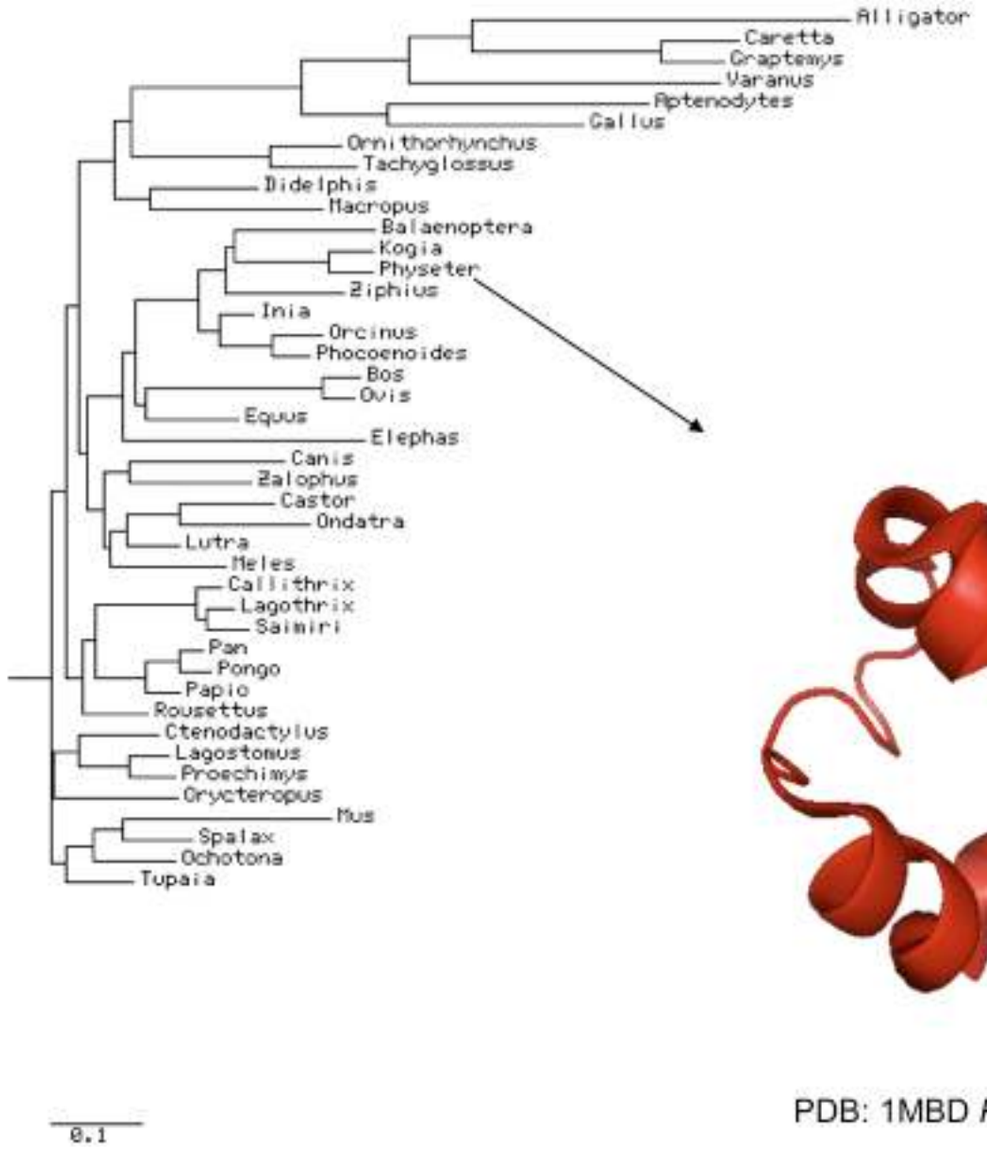
1PMB

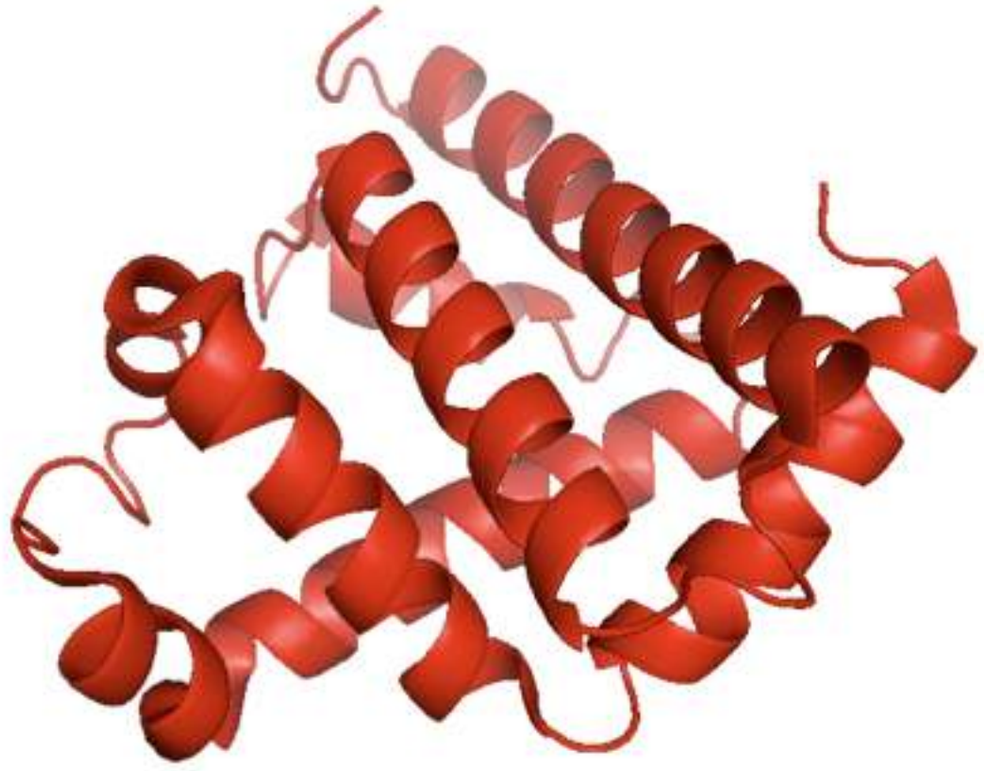
(i,i+4): (0,4), (1,5), (2,6), ...

(i,i+3): (0,3), (1,4), (2,5), ...

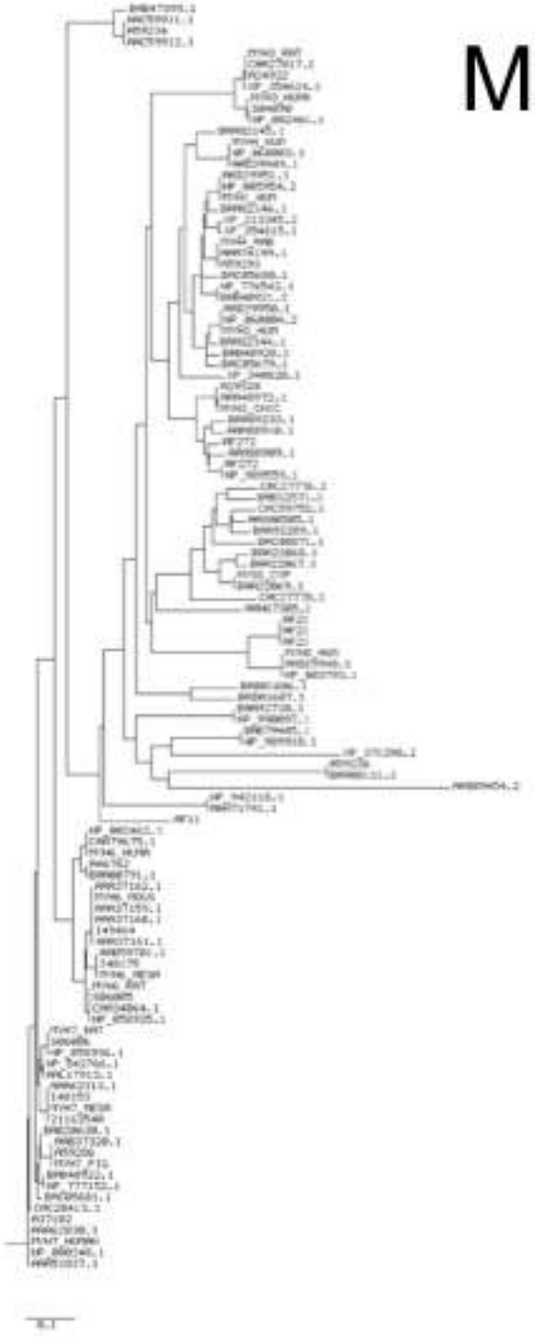
Myoglobin

42 Tetrapod sequences
153 residues,
mainly alpha (~70%)

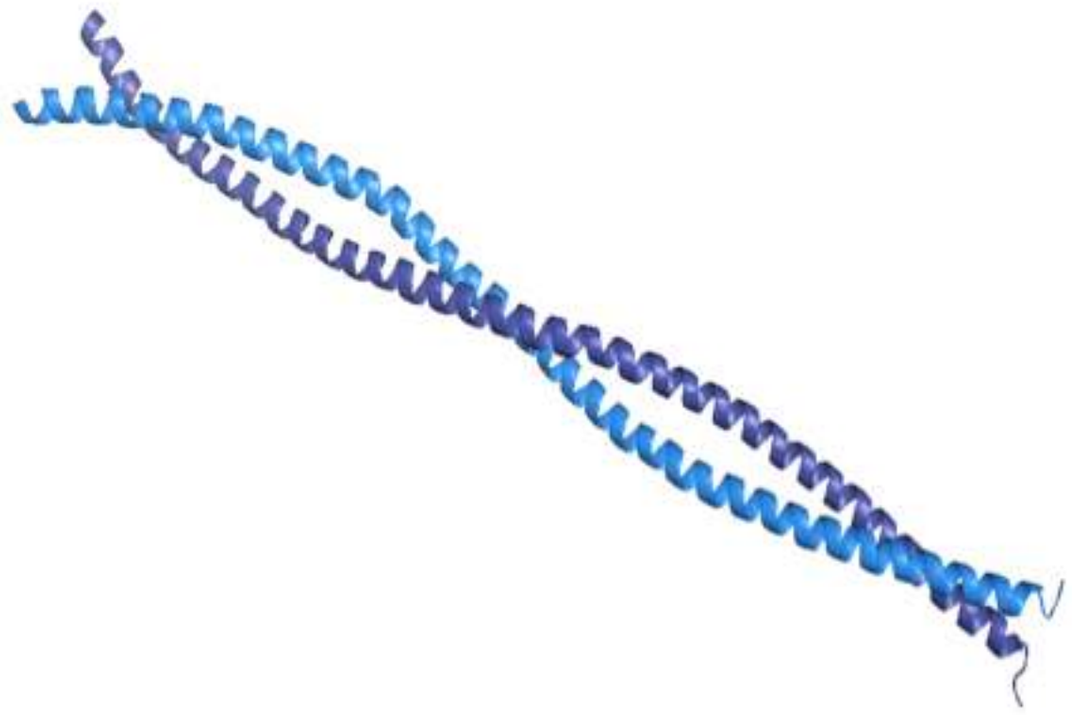




PDB: 1MBD *Physeter macrocephalus* (sperm whale) myoglobin

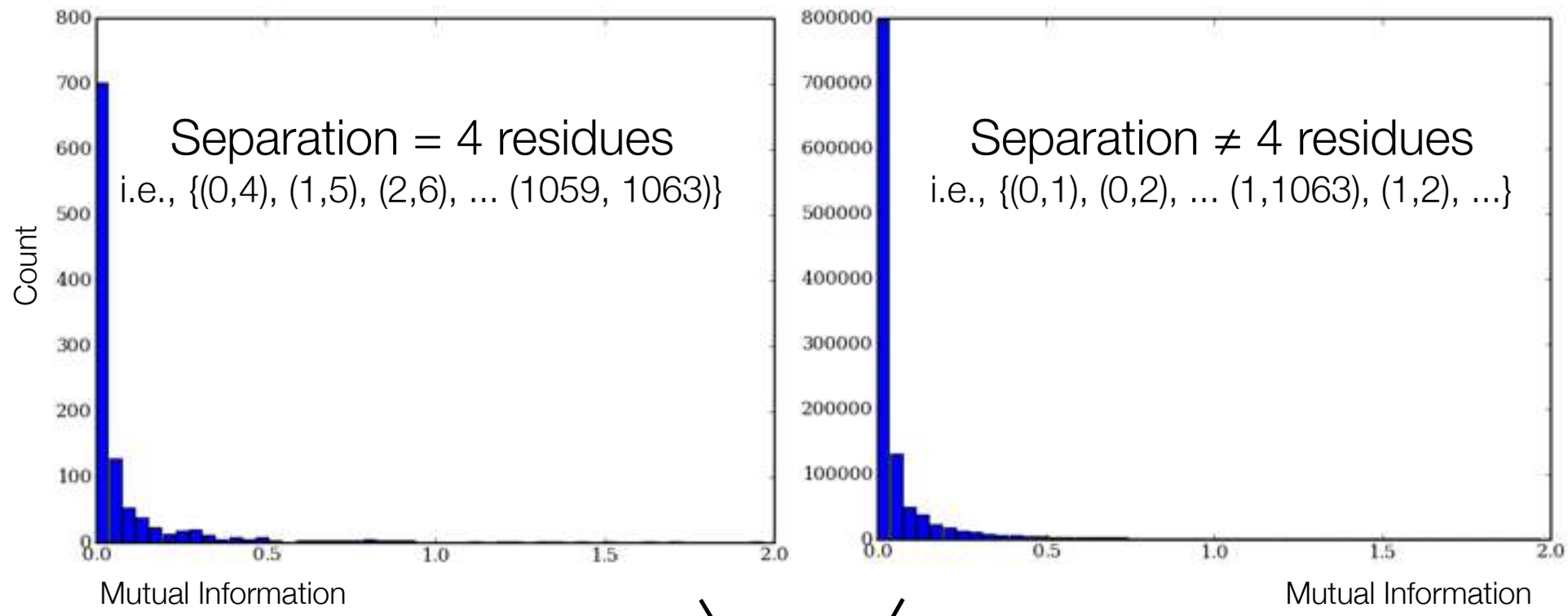


Myosin Heavy Chain (MyHC)
Rod Domain
114 Chordate sequences
1064 residue alpha helix



Does the distribution of coevolution scores corresponding to stacked residues differ from distribution of all other scores?

Coevolution matrix



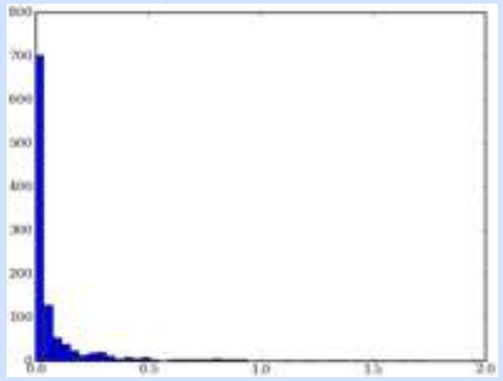
One-tailed, two-sample t-test

(confirmed with Wilcoxon rank-sum and matrix permutation test)

Tree-ignorant	**Tree-aware**
Mutual Information (MI)	Ancestral States (AS)
Normalized Mutual Information (NMI)	Generalized Continuous-Time Markov Process Coevolutionary Algorithm (GCTMPCA or G)
Statistical Coupling Analysis (SCA)	LnLCorr, two implementations (LnLCorr99 and LnLCorr07 or L99 and L07)
Resampled Mutual Information (RMI)	
Corrected Mutation Information (MIp)	CoMap (CM)

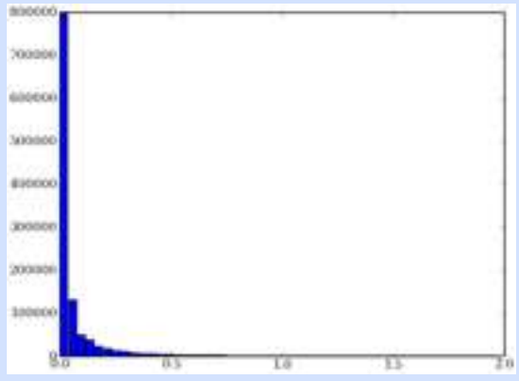
Separation = 4 residues

i.e., {(0,4), (1,5), (2,6), ... (1059, 1063)}



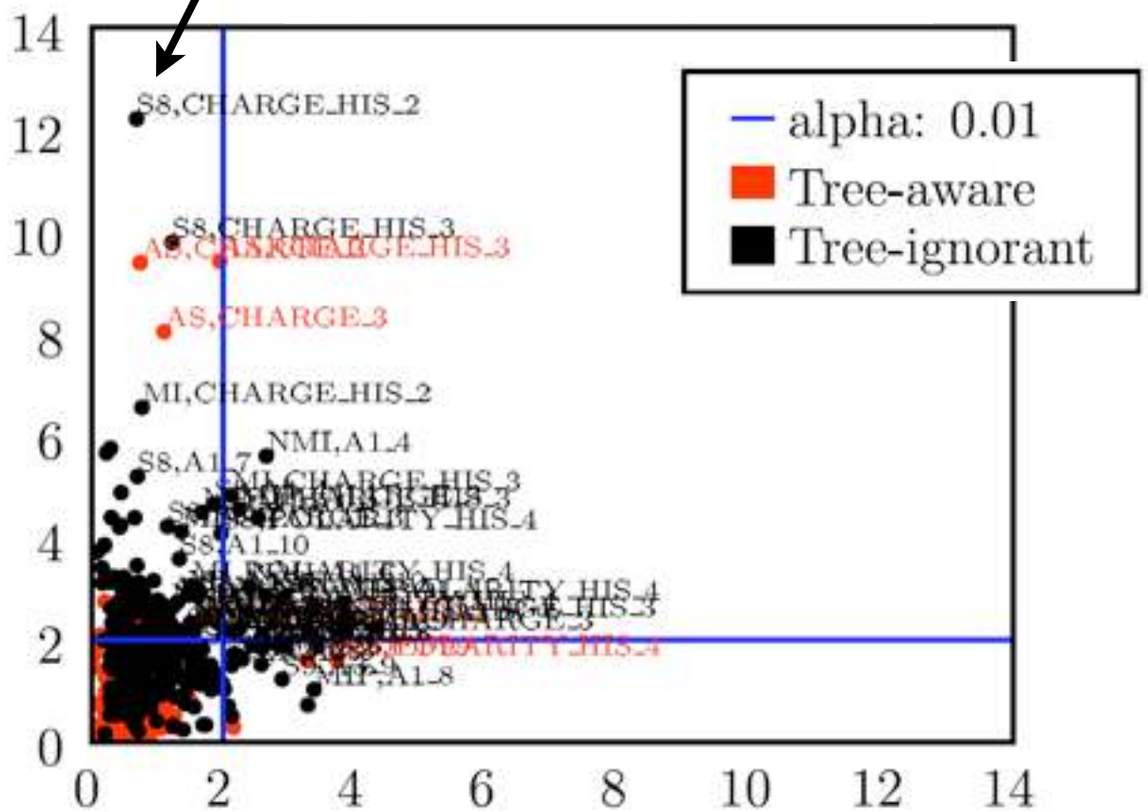
Separation ≠ 4 residues

i.e., {(0,1), (0,2), ... (1,1063), (1,2), ...}



One-tailed, two-sample t-test

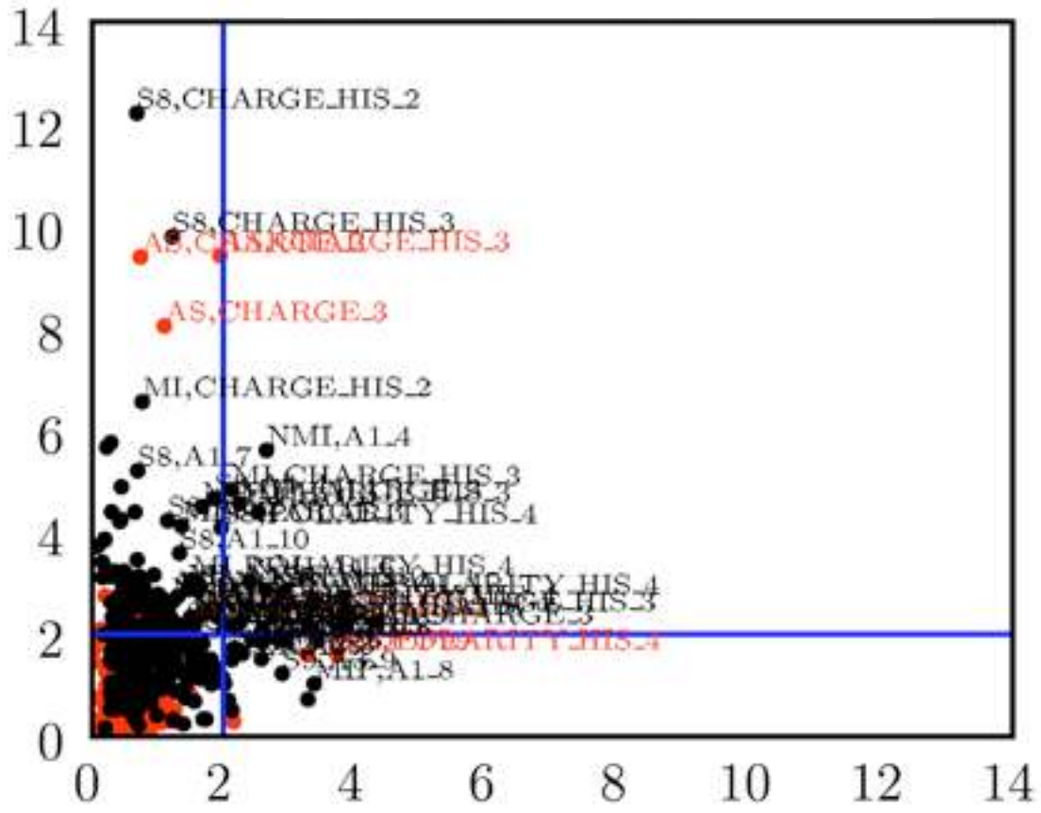
Each point represents a (method,alphabet) combination



y-axis:
-log(P-value)
for (i,i+4) pairs

x-axis:
-log(P-value) for (i,i+3) pairs

-log(P-value)
for (i,i+4) pairs



-log(P-value) for (i,i+3) pairs

1: identifies (i, i+4) pairs
2: identifies (i, i+3) and (i, i+4) pairs
3: identifies neither
4: identifies (i, i+3) pairs

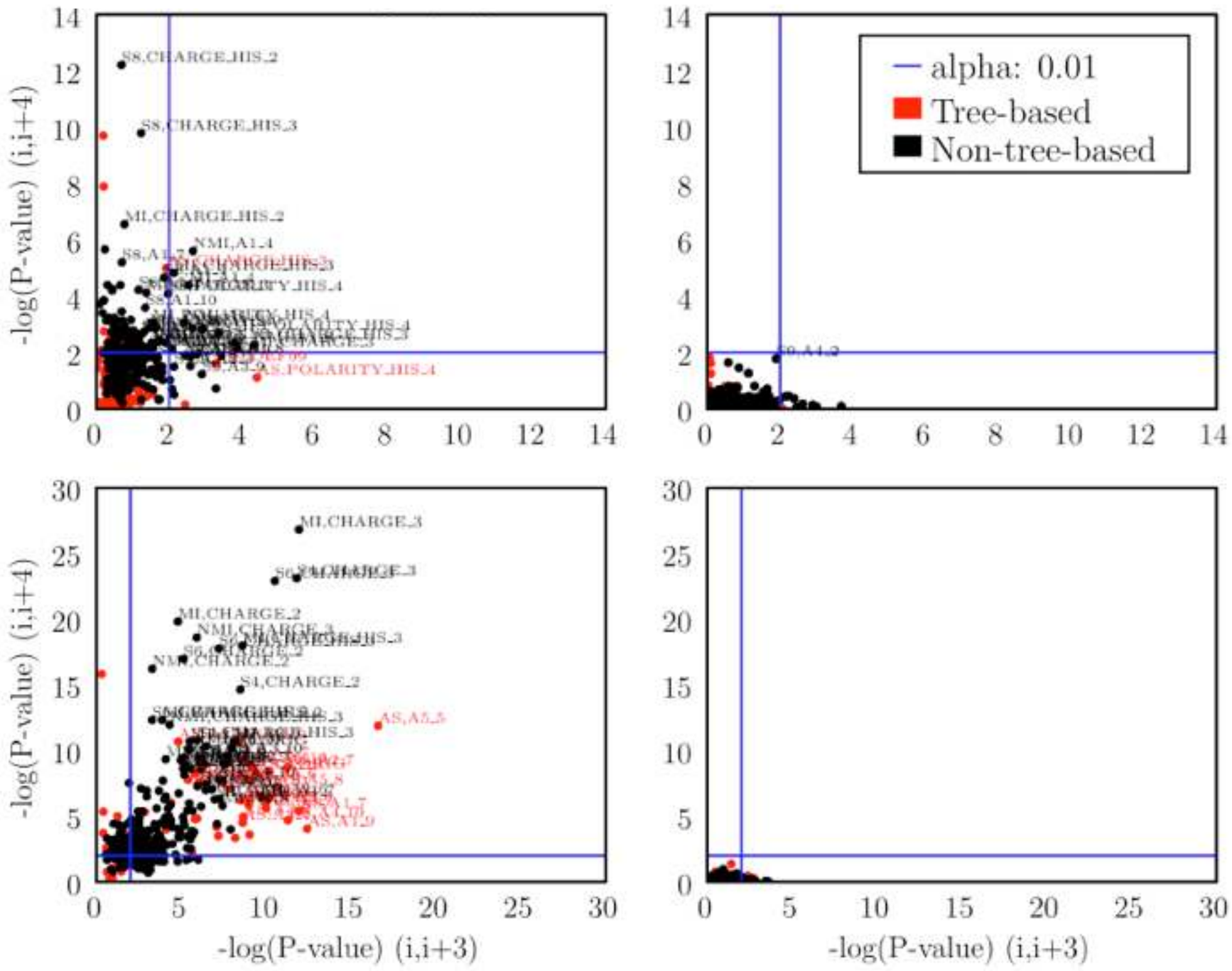
Tree-aware versus tree-ignorant comparison

Positive control

Negative control

Myoglobin

Myosin rod



Tree-aware versus tree-ignorant comparison

(A) Tetrapod Myoglobin						
	Tree-aware		Tree-ignorant			
	$p \leq \alpha$	$p > \alpha$	$p \leq \alpha$	$p > \alpha$	χ^2	$p - value$
(i,i+3) w best SCA (0.9)	3	116	21	191	6.18	1.29e-02
(i,i+3) w worst SCA (0.5)	3	116	12	200	1.74	1.88e-01
(i,i+4) w best SCA (0.8)	13	106	**110**	**102**	**54.77**	**1.36e-13**
(i,i+4) w worst SCA (0.4)	13	106	**82**	**130**	**28.69**	**8.49e-08**
(B) Randomized Tetrapod Myoglobin						
	Tree-aware		Tree-ignorant			
	$p \leq \alpha$	$p > \alpha$	$p \leq \alpha$	$p > \alpha$	χ^2	$p - value$
(i,i+3) w best SCA (0.9)	1	118	4	208	0.56	4.54e-01
(i,i+3) w worst SCA (0.5)	1	118	9	203	3.02	8.24e-02
(i,i+4) w best SCA (0.8)	0	119	0	212	n/a	n/a
(i,i+4) w worst SCA (0.4)	0	119	0	212	n/a	n/a
(C) Chordate Myosin						
	Tree-aware		Tree-ignorant			
	$p \leq \alpha$	$p > \alpha$	$p \leq \alpha$	$p > \alpha$	χ^2	$p - value$
(i,i+3) w best SCA (0.4)	43	16	175	37	2.74	9.78e-02
(i,i+3) w worst SCA (0.9)	43	16	152	60	0.03	8.58e-01
(i,i+4) w best SCA (0.6)	47	12	166	46	0.05	8.22e-01
(i,i+4) w worst SCA (0.9)	47	12	134	78	5.63	1.76e-02
(D) Randomized Chordate Myosin						
	Tree-aware		Tree-ignorant			
	$p \leq \alpha$	$p > \alpha$	$p \leq \alpha$	$p > \alpha$	χ^2	$p - value$
(i,i+3) w best SCA (0.4)	6	54	29	183	0.56	4.52e-01
(i,i+3) w worst SCA (0.9)	6	54	28	184	0.44	5.07e-01
(i,i+4) w best SCA (0.6)	0	60	0	212	n/a	n/a
(i,i+4) w worst SCA (0.9)	0	60	0	212	n/a	n/a

Alphabet type: recoding based on appropriate physicochemical property improves performance?

(a) Tetrapod Myoglobin								
Rank	MI	NMI	RMI	SCA8	GCTMPCA7	LnLCorr99*	LnLCorr07*	AS
1	CHARGE_HIS_2	A1_4	A3_2	CHARGE_HIS_2	A4_7	DEF99	DEF99	CHARGE_2
2	CHARGE_2	A1_3	A3_4	CHARGE_HIS_3	A1_10	A1	A1	CHARGE_3
3	CHARGE_HIS_3	A1_7	A4_4	A1_7	A4_10	A3	DEF07	CHARGE_HIS_3
4	A1_4	CHARGE_HIS_2	A5_10	A1_4	A5_5	A4	A2	CHARGE_HIS_2
5	CHARGE_3	CHARGE_2	A4_7	A1_2	A5_10	A2	A4	POLARITY_HIS_4
(b) Chordate Myosin								
Rank	MI	NMI	RMI	SCA6	GCTMPCA7*	LnLCorr99	LnLCorr07*	AS
1	CHARGE_3	CHARGE_3	A4_3	CHARGE_3	-	DEF99	-	A1_2
2	CHARGE_2	CHARGE_2	A4_4	CHARGE_HIS_3	-	A4	-	A5_5
3	CHARGE_HIS_3	CHARGE_HIS_3	A3_3	CHARGE_2	-	A5	-	A2_5
4	CHARGE_HIS_2	CHARGE_HIS_2	A3_4	CHARGE_HIS_2	-	A2	-	A5_9
5	A3_9	A4_4	CHARGE_2	SIZE_2	-	A1	-	A5_7

Comparison conclusions

Tree-ignorant methods can match or out-perform tree-aware methods, although results *do not* imply that controlling for phylogeny is unimportant.

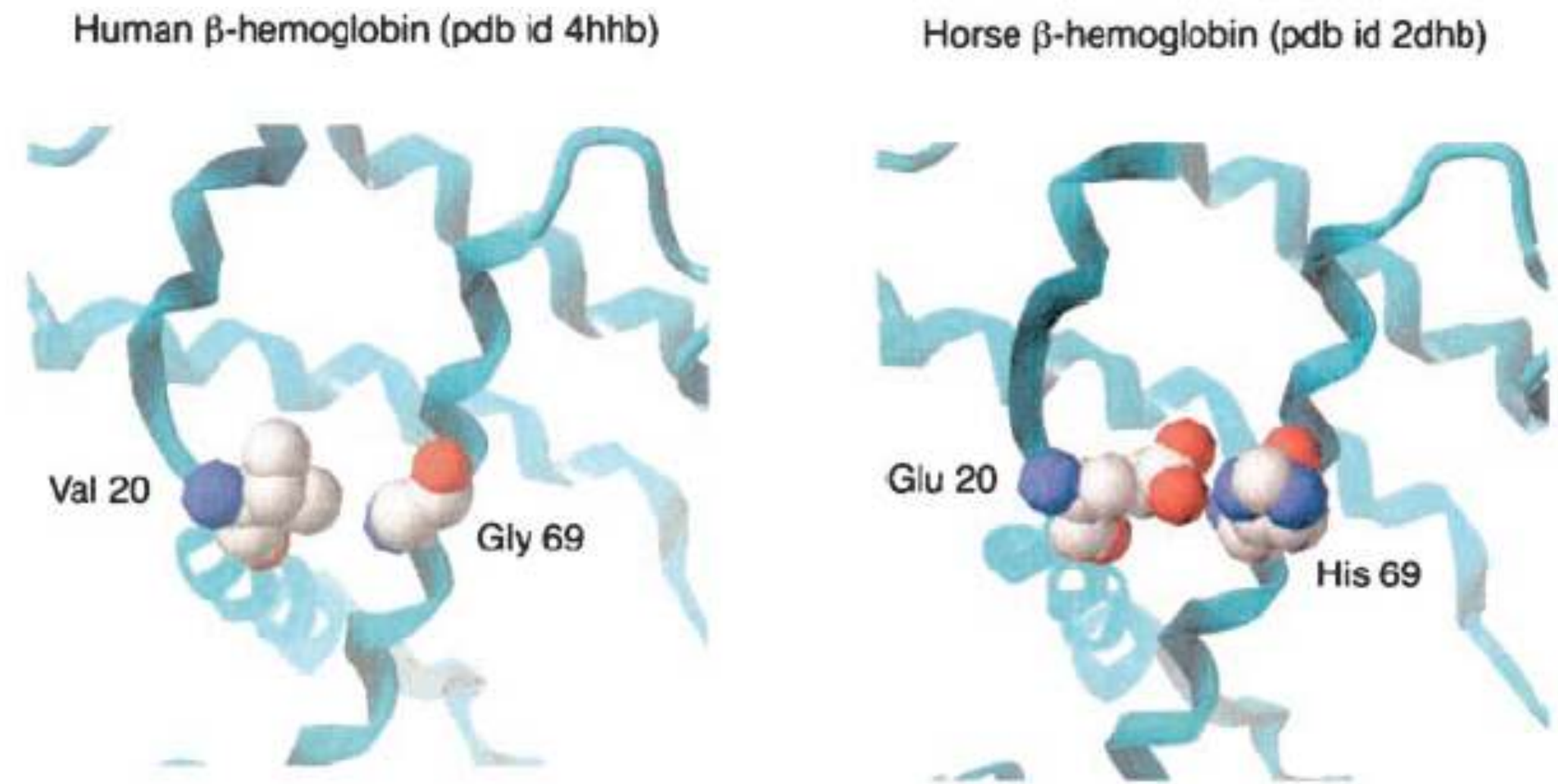
Charge-based alphabets exhibit best performance for detecting coevolution resulting from residue stacking in alpha helices.

Detecting coevolution without phylogenetic trees? Tree-ignorant metrics of coevolution perform as well as tree-aware metrics. J. Gregory Caporaso, Sandra Smit, Brett C. Easton, Lawrence Hunter, Gavin A. Huttley, and Rob Knight. BMC Evolutionary Biology, December, 2008.

Should covariation patterns arising from coevolution be expected to mirror phylogeny?

Do interactions frequently involve different positions in different sequences, posing a problem for all pairwise coevolution algorithms?

Compensatory relationships should be detectable using coevolution algorithms.

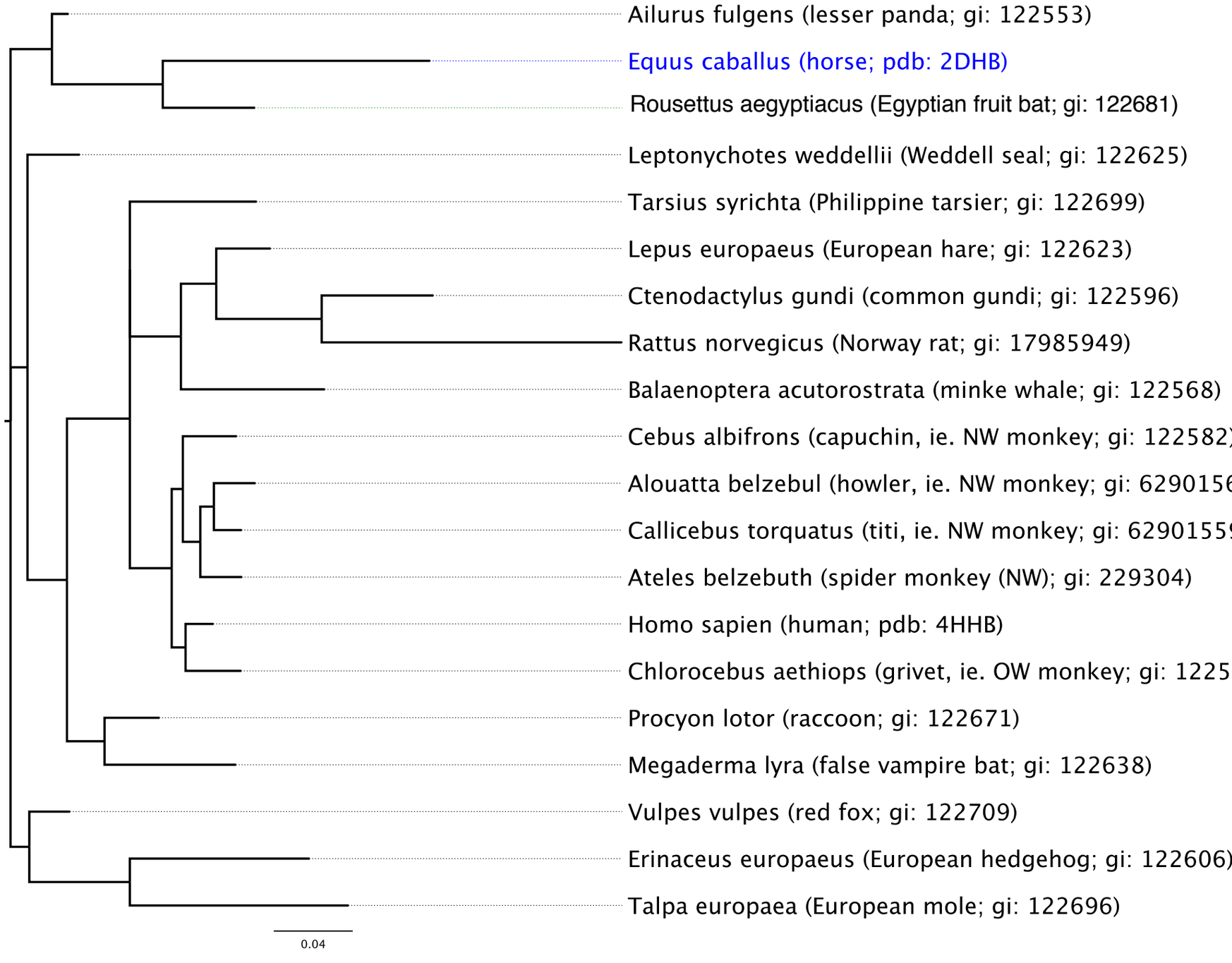


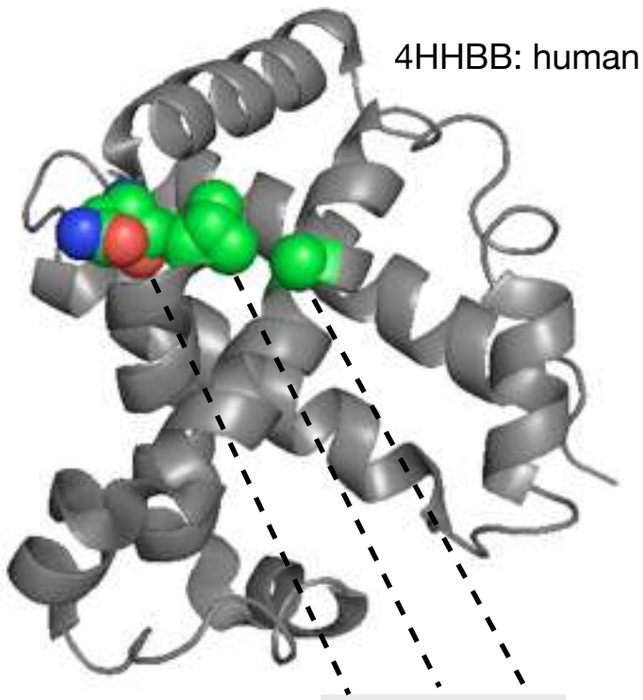
Val20Glu is pathogenic in humans, but a second mutation is present which restores function.

Figure modified from Kondrashov et al. (2002)

E20/H69 is monophyletic

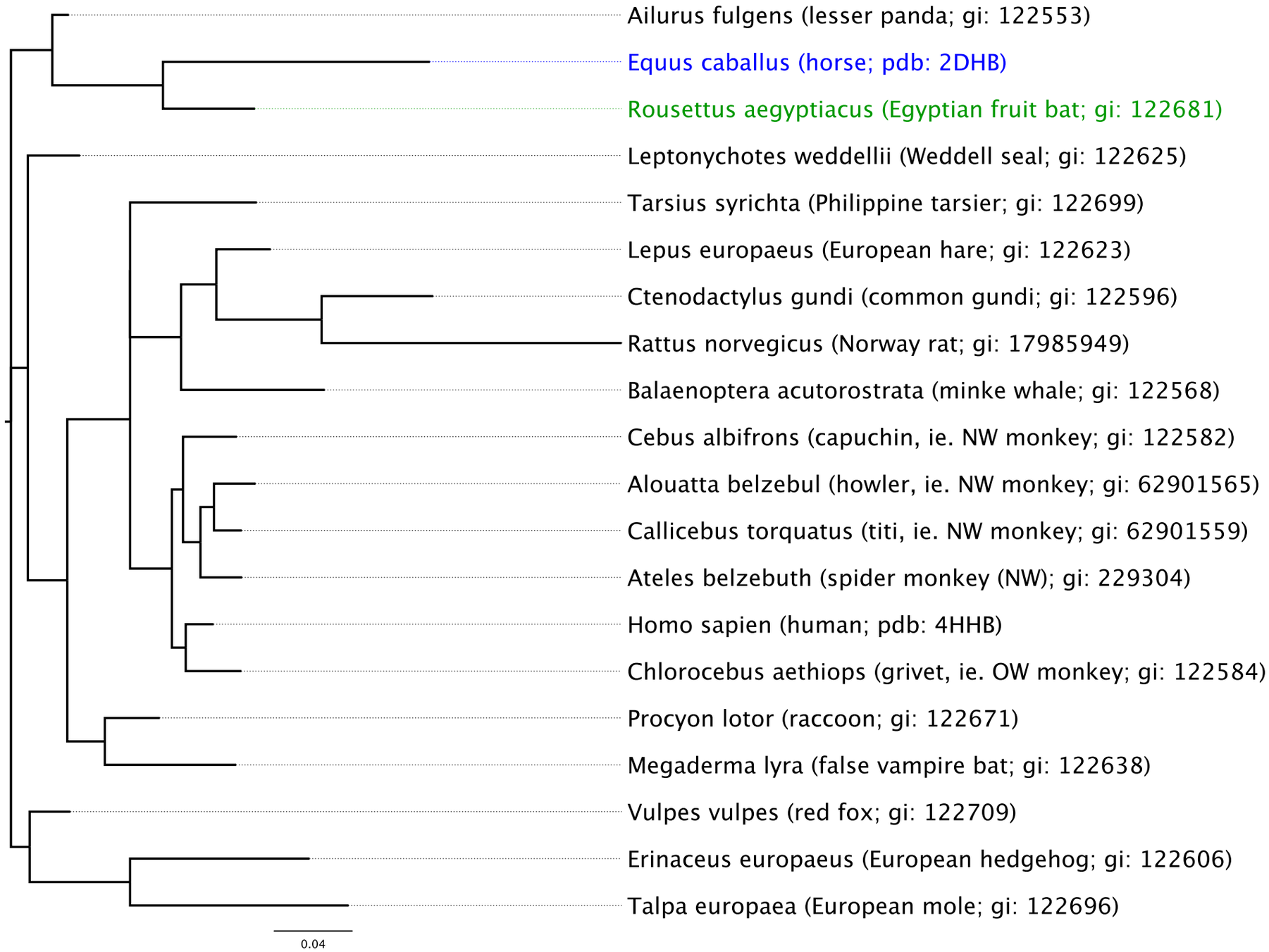
	Position	
	20	69
Ailurus	V	N
Equus	E	H
Rousettus	V	D
Leptonychotes	V	N
Tarsius	V	N
Lepus	V	A
Ctenodactylus	V	T
Rattus	P	N
Balaenoptera	V	A
Cebus	V	G
Alouatta	V	G
Callicebus	V	G
Ateles	V	G
Homo (structure above)	V	G
Chlorocebus	V	G
Procyon	V	N
Megaderma	V	N
Vulpes	V	N
Erinaceus	V	Q
Talpa	V	N





4HHBB: human beta hemoglobin

	Position		
	19	20	69
Ailurus	N	V	N
Equus	N	E*	H
Rousettus	K	V	D*
Leptonychotes	N	V	N
Tarsius	D	V	N
Lepus	N	V	A
Ctenodactylus	N	V	T
Rattus	N	P	N
Balaenoptera	N	V	A
Cebus	N	V	G
Alouatta	N	V	G
Callicebus	N	V	G
Ateles	N	V	G
Homo (structure above)	N	V	G
Chlorocebus	N	V	G
Procyon	N	V	N
Megaderma	N	V	N
Vulpes	N	V	N
Erinaceus	K	V	Q
Talpa	N	V	N



* Starred residues are associated with pathogenicity in human beta hemoglobin at the corresponding position.

Sequence Co-occurrence and Covariation Suggest a Model for the Type VI Secretion System

---

Identify the conserved pairwise protein-protein interactions in functional Type VI Secretion System complexes.

Test the hypothesis that sequence covariation can suggest pairs of proteins that are likely to be involved in pairwise interactions.

Image by Emily Hoyer

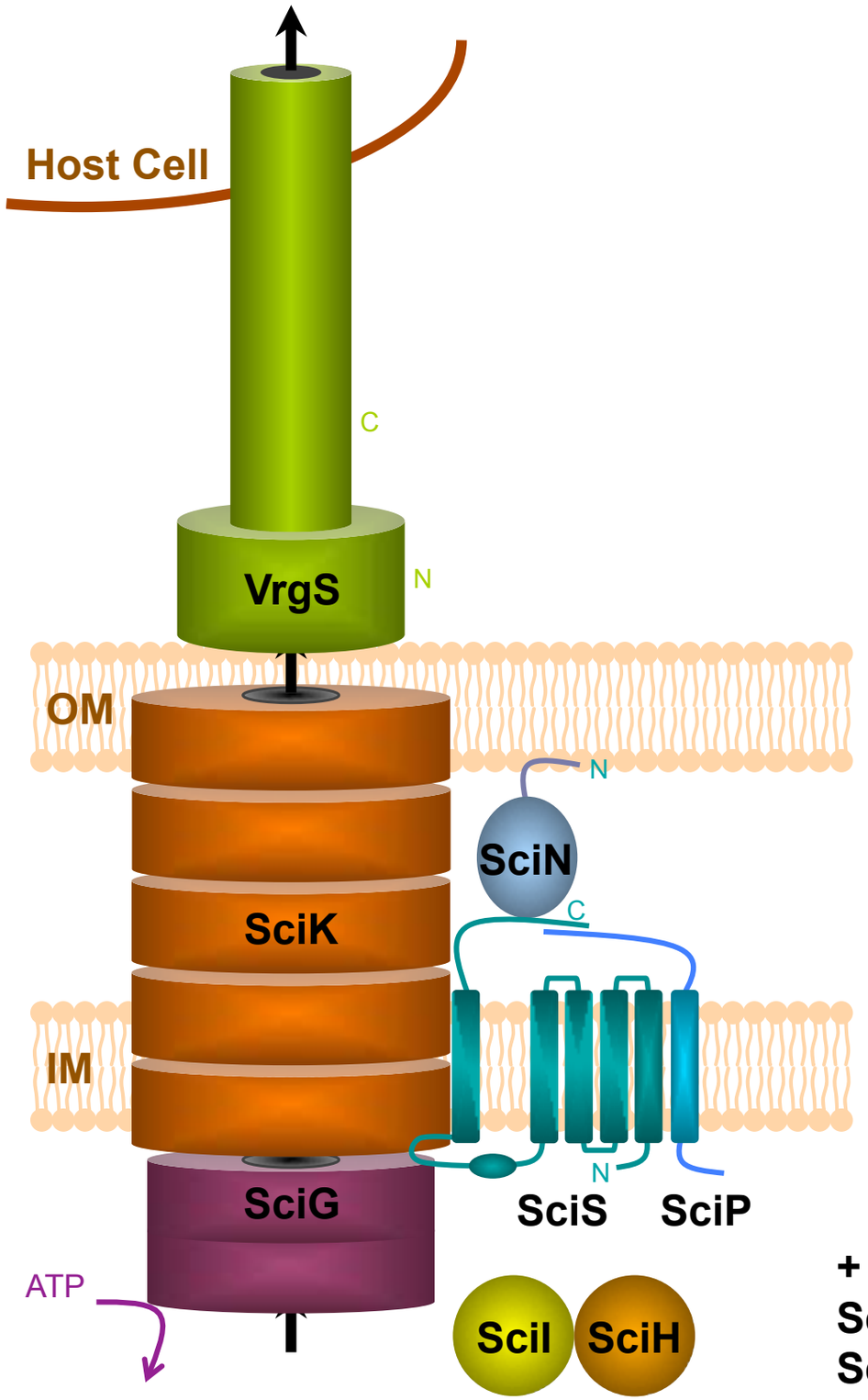
Type VI Secretion System

First evidence ~1996
Named in 2006

Implicated in pathogenicity of many species including V. cholerae, Samonella enterica

>20 proteins suspected

Little known about the interactions



+ cytoplasmic proteins:
SciA, SciB, SciC,
SciD, SciE & SciO

Image credit: Amy Dear, based on literature review

	Protein A													Protein B								
Species/Position	1	.	.	.	5	.	.	.	.	1	0	.	.	1	.	.	.	5	.	.	.	.
a	P	R	N	I	K	D	K	P	-	E	A	H	F	Y	T	T	I	L	K	H	F	F
b	P	R	N	I	K	D	K	P	-	E	A	H	F	Y	T	F	L	L	K	H	F	F
c	P	R	N	I	K	D	K	P	-	E	A	H	F	Y	S	T	I	L	R	H	F	F
d	P	K	P	A	K	R	K	A	-	E	A	H	F	Y	T	T	I	L	E	H	W	F
e	P	R	P	P	K	R	K	Q	A	E	A	H	L	Y	S	T	I	L	E	H	F	F
f	P	K	P	A	K	R	K	V	-	E	A	H	F	Y	T	T	I	L	E	H	F	F

	Protein A													Protein B								
Species/Position	1	.	.	.	5	.	.	.	.	1	0	.	.	1	.	.	.	5	.	.	.	.
a	P	R	N	I	K	D	K	P	-	E	A	H	F	Y	T	T	I	L	K	H	F	F
b	P	R	N	I	K	D	K	P	-	E	A	H	F	Y	T	F	L	L	K	H	F	F
c	P	R	N	I	K	D	K	P	-	E	A	H	F	Y	S	T	I	L	R	H	F	F
d	P	K	P	A	K	R	K	A	-	E	A	H	F	Y	T	T	I	L	E	H	W	F
e	P	R	P	P	K	R	K	Q	A	E	A	H	L	Y	S	T	I	L	E	H	F	F
f	P	K	P	A	K	R	K	V	-	E	A	H	F	Y	T	T	I	L	E	H	F	F

Compile alignments from seed sequences

Associate sequences expected to form functional T6SS complex: the sequence pairing problem

Apply covariation analysis and post-processing steps

Identify highest scoring pairs

Counts of gene cooccurrence within a gene cluster in bacterial genomes.

	SciA	SciB	SciC	SciD	SciE	SciF	SciG	SciH	SciI	SciK	SciM	SciN	SciO	SciP	SciS	SciT	SciU	SciV	SciW	STM0291	VrgS
SciA	38																				
SciB	37	68																			
SciC	38	68	70																		
SciD	34	43	44	44																	
SciE	16	22	23	22	23																
SciF	7	6	7	7	7	7															
SciG	35	60	60	40	21	6	61														
SciH	37	64	66	43	23	7	57	67													
SciI	37	65	67	43	22	6	58	66	68												
SciK	35	45	47	42	23	7	41	46	46	47											
SciM	13	12	13	11	8	6	11	13	12	13	13										
SciN	28	50	52	30	17	7	49	49	50	32	10	53									
SciO	38	67	69	43	23	7	61	66	67	46	13	53	70								
SciP	38	67	69	44	23	7	61	66	67	47	13	53	69	70							
SciS	28	52	53	36	20	5	48	51	52	39	8	44	52	53	53						
SciT	10	15	15	13	10	2	14	15	15	14	2	12	15	15	15	15					
SciU	4	4	4	4	4	2	4	4	3	4	2	4	4	4	3	3	4				
SciV	4	4	4	4	4	2	4	4	3	4	2	4	4	4	3	3	4	4			
SciW	2	5	5	2	2	2	5	4	3	2	2	5	5	5	4	1	2	2	5		
STM0291	1	4	4	1	1	1	4	3	2	1	1	4	4	4	3	0	1	1	4	4	
VrgS	32	56	57	39	20	7	51	55	56	40	13	44	57	57	44	12	3	3	5	4	58

count < 30
30 >= *count* > 40
count >= 40
individual gene count

	Protein A													Protein B								
Species/Position	1	.	.	.	5	.	.	.	.	1	0	.	.	1	.	.	.	5	.	.	.	.
a	P	R	N	I	K	D	K	P	-	E	A	H	F	Y	T	T	I	L	K	H	F	F
b	P	R	N	I	K	D	K	P	-	E	A	H	F	Y	T	F	L	L	K	H	F	F
c	P	R	N	I	K	D	K	P	-	E	A	H	F	Y	S	T	I	L	R	H	F	F
d	P	K	P	A	K	R	K	A	-	E	A	H	F	Y	T	T	I	L	E	H	W	F
e	P	R	P	P	K	R	K	Q	A	E	A	H	L	Y	S	T	I	L	E	H	F	F
f	P	K	P	A	K	R	K	V	-	E	A	H	F	Y	T	T	I	L	E	H	F	F

Joint-entropy-normalized Mutual Information (NMI)

Charge and polarity-based reduced state alphabets (in addition to full alphabet)

Filters to exclude apparent stochastic variation

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	**nan**	0.00	0.00	0.00	nan	0.00	nan	0.00	nan	nan	nan	nan	0.00
2	0.00	0.16	0.00	0.24	0.00	0.00	0.00	0.20	nan	0.00	0.00	0.00	0.25
3	0.00	0.07	0.13	**0.10**	0.00	0.13	0.00	0.08	nan	0.00	0.00	0.00	0.04
4	0.00	0.07	0.13	0.10	0.00	0.13	0.00	0.08	nan	0.00	0.00	0.00	0.04
5	nan	0.00	0.00	0.00	nan	0.00	nan	0.00	nan	nan	nan	nan	0.00
6	0.00	0.24	0.69	0.52	0.00	**0.69**	0.00	0.44	nan	0.00	0.00	0.00	0.10
7	nan	0.00	0.00	0.00	nan	0.00	nan	0.00	nan	nan	nan	nan	0.00
8	0.00	0.25	0.13	0.18	0.00	0.13	0.00	0.36	nan	0.00	0.00	0.00	0.04
9	nan	0.00	0.00	0.00	nan	0.00	nan	0.00	nan	nan	nan	nan	0.00

	SciG and SciK: predicted to interact			SciD and SciP: not predicted to interact		
NMI	Bin count	count $\geq$ NMI	percent $\geq$ NMI	Bin count	count $\geq$ NMI	percent $\geq$ NMI
0.00	23854	28982	1.0e+00	9170	11049	1.0e+00
0.05	4012	5128	1.8e-01	1508	1879	1.7e-01
0.10	760	1116	3.9e-02	295	371	3.4e-02
0.15	221	356	1.2e-02	51	76	6.9e-03
0.20	77	135	4.7e-03	15	25	2.3e-03
0.25	12	58	2.0e-03	6	10	9.1e-04
0.30	6	46	1.6e-03	1	4	3.6e-04
0.35	6	40	1.4e-03	1	3	2.7e-04
0.40	9	34	1.2e-03	1	2	1.8e-04
0.45	6	25	8.6e-04	1	1	9.1e-05
0.50	0	19	6.6e-04	0	0	0.0e+00
0.55	0	19	6.6e-04	0	0	0.0e+00
0.60	0	19	6.6e-04	0	0	0.0e+00
0.65	2	19	6.6e-04	0	0	0.0e+00
0.70	0	17	5.9e-04	0	0	0.0e+00
0.75	0	17	5.9e-04	0	0	0.0e+00
0.80	5	17	5.9e-04	0	0	0.0e+00
0.85	0	12	4.1e-04	0	0	0.0e+00
0.90	0	12	4.1e-04	0	0	0.0e+00
0.95	12	12	4.1e-04	0	0	0.0e+00

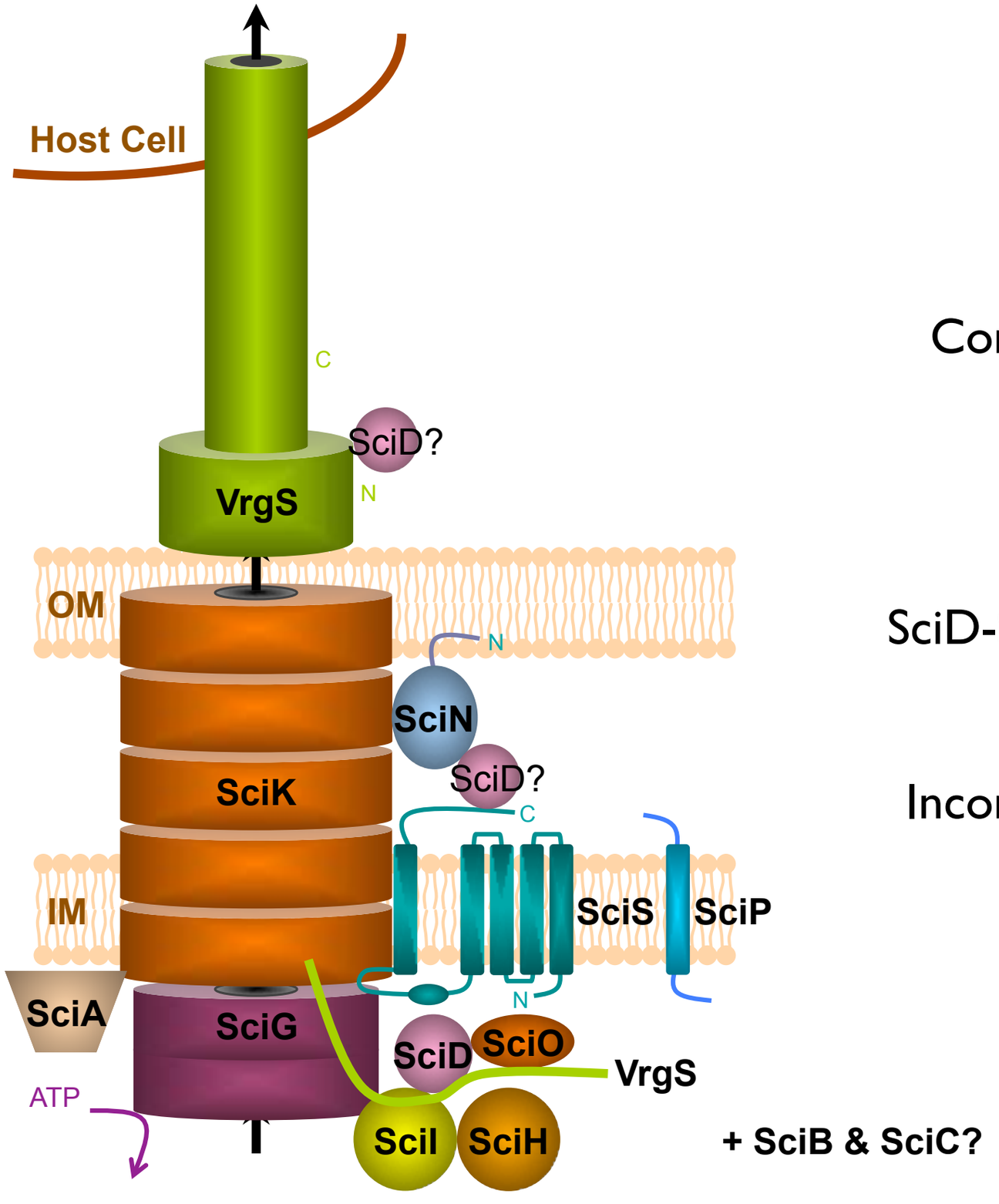


Image credit: Amy Dear, based on literature review and covariation-based predictions

Sixteen sequence covariation-based predictions

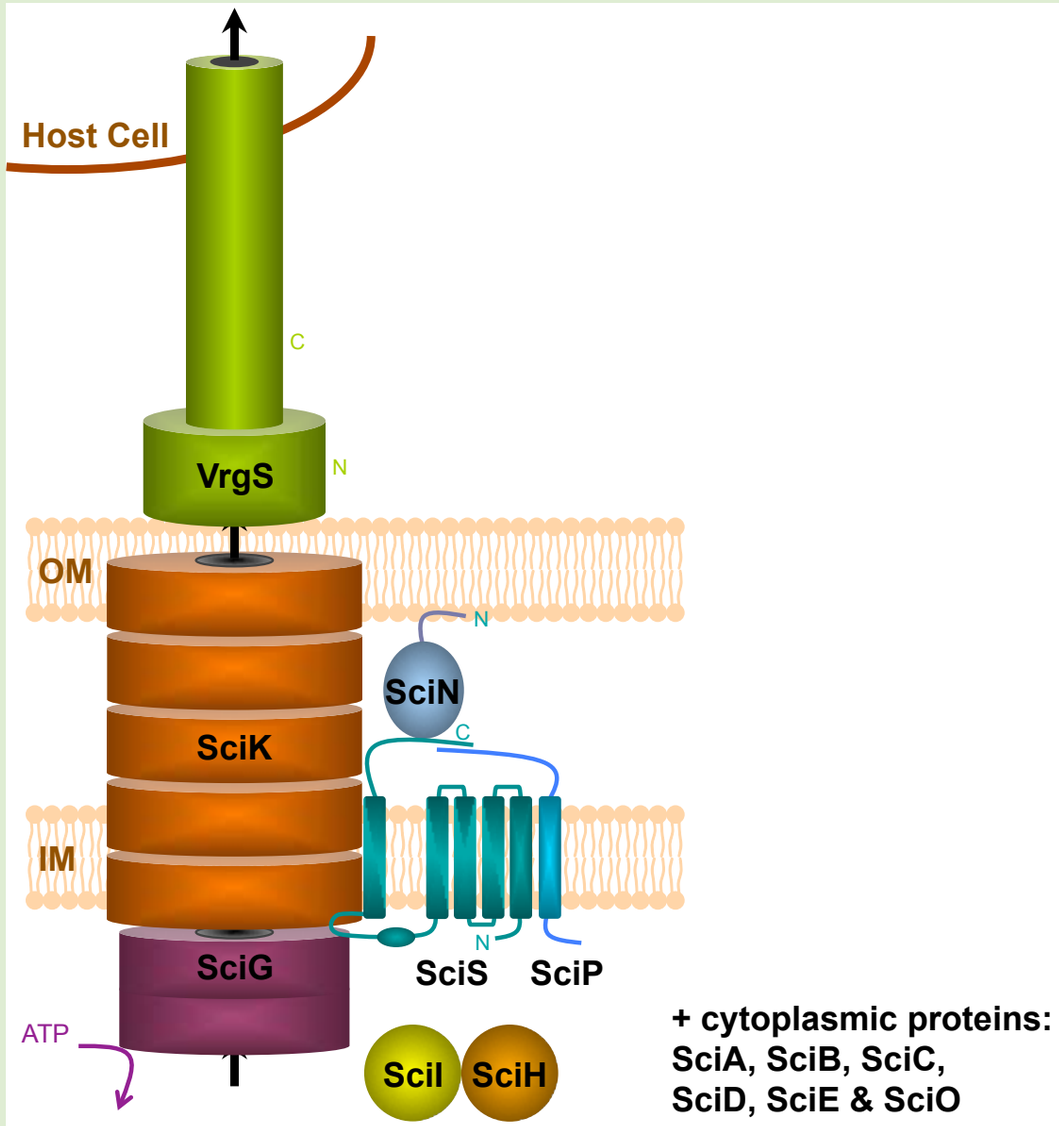
Confirmed biochemically (2 of 16)
SciH-SciI; SciG-SciI

Consistent with current information (5 of 16)
SciG-SciK; SciK-SciN; SciK-SciS; SciG-SciS; SciK-VrgS

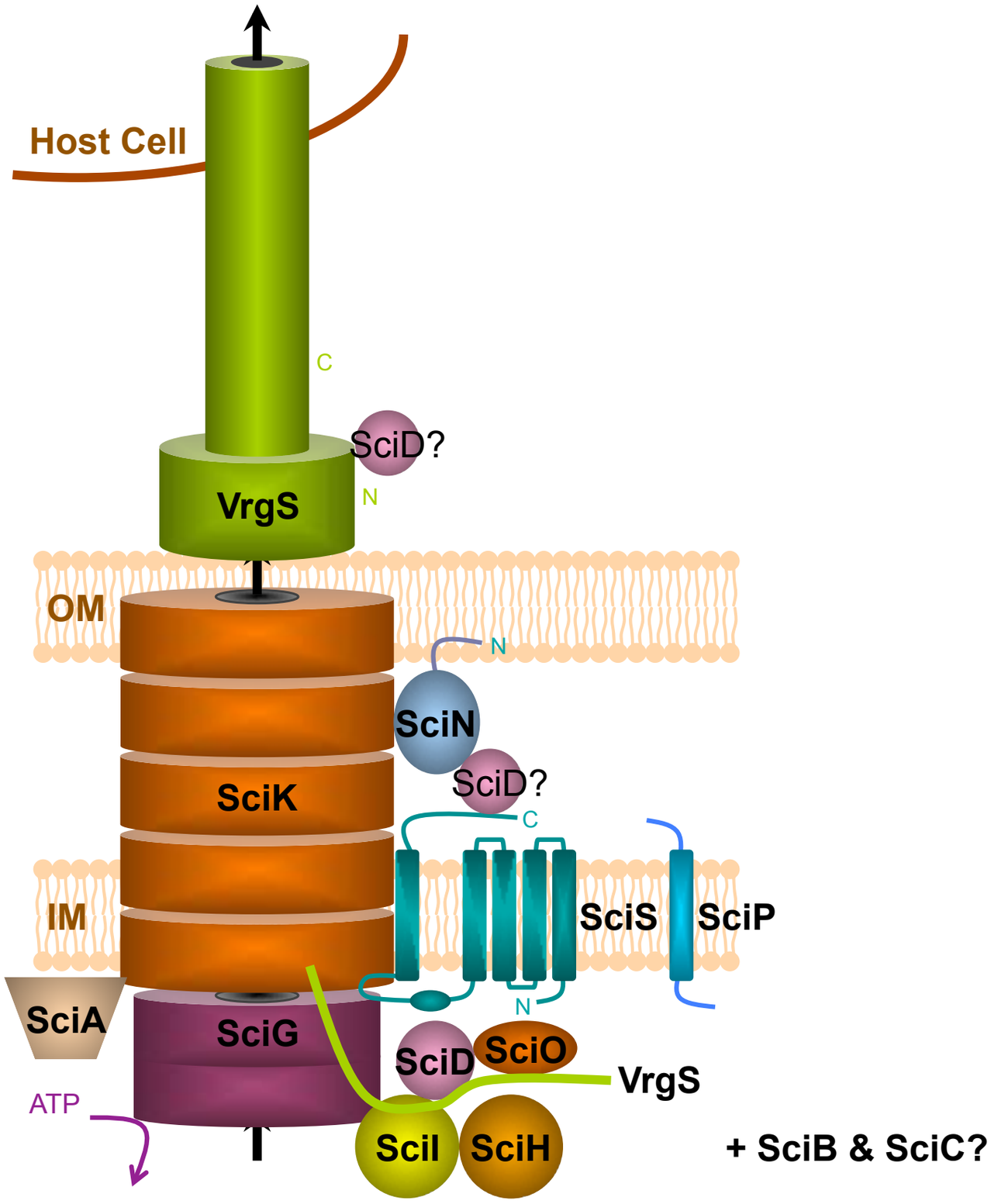
Suspected cytoplasmic (8 of 16)
SciD-SciO; SciI-VrgS; SciD-SciS; SciD-VrgS; SciD-SciI; SciA-SciK; SciO-VrgS; SciG-VrgS

Inconsistent with current information (1 of 16)
SciD-SciN

Prior model



+ cytoplasmic proteins:
SciA, SciB, SciC,
SciD, SciE & SciO



+ SciB & SciC?

Summary of PyCogent features that facilitate coevolution analysis

Application controllers: LnLCorr, GCTMPCA, ...

Stats module: t-test, Wilcoxon Rank Sum, chi-squared/ G-test

Core bioinformatics objects: tree, alignment (filtering particularly useful)

Parsers: alignment and tree file formats

Software integrated into PyCogent

Coevolution module: open source implementations or wrappers for seven of the coevolution algorithms

Coevolution script: easy access to these methods with common interface

Amino acid alphabet definitions and reduction code

Writing PyCogent scripts

http://bit.ly/mPaQjN

Learning more about PyCogent

- Python: http://www.python.org
- Documentation: http://pycogent.sourceforge.net
- Install: http://pycogent.sourceforge.net/install.html
- If you're interested in contributing to PyCogent you can contact me at gregcaporaso@gmail.com. (Contributions are peer-reviewed by developers before developer access is granted.) These pages are for you:
  - http://pycogent.sourceforge.net/coding_guidelines.html
  - http://pycogent.sourceforge.net/developer_notes.html

Defining a new application controller: formatdb

This work was funded by the HMP DACC (www.hmpdacc.org) and by the Bill and Melinda Gates Foundation.