Estimation of abundances in microbial communities from metagenomic data

September 6th 2018
Master 2 Informatique Fondamentale
Université Paris-Est Marne-la-Vallée

Simone Pignotti

UPEM UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE

Supervised at LIGM by:
Gregory Kucherov

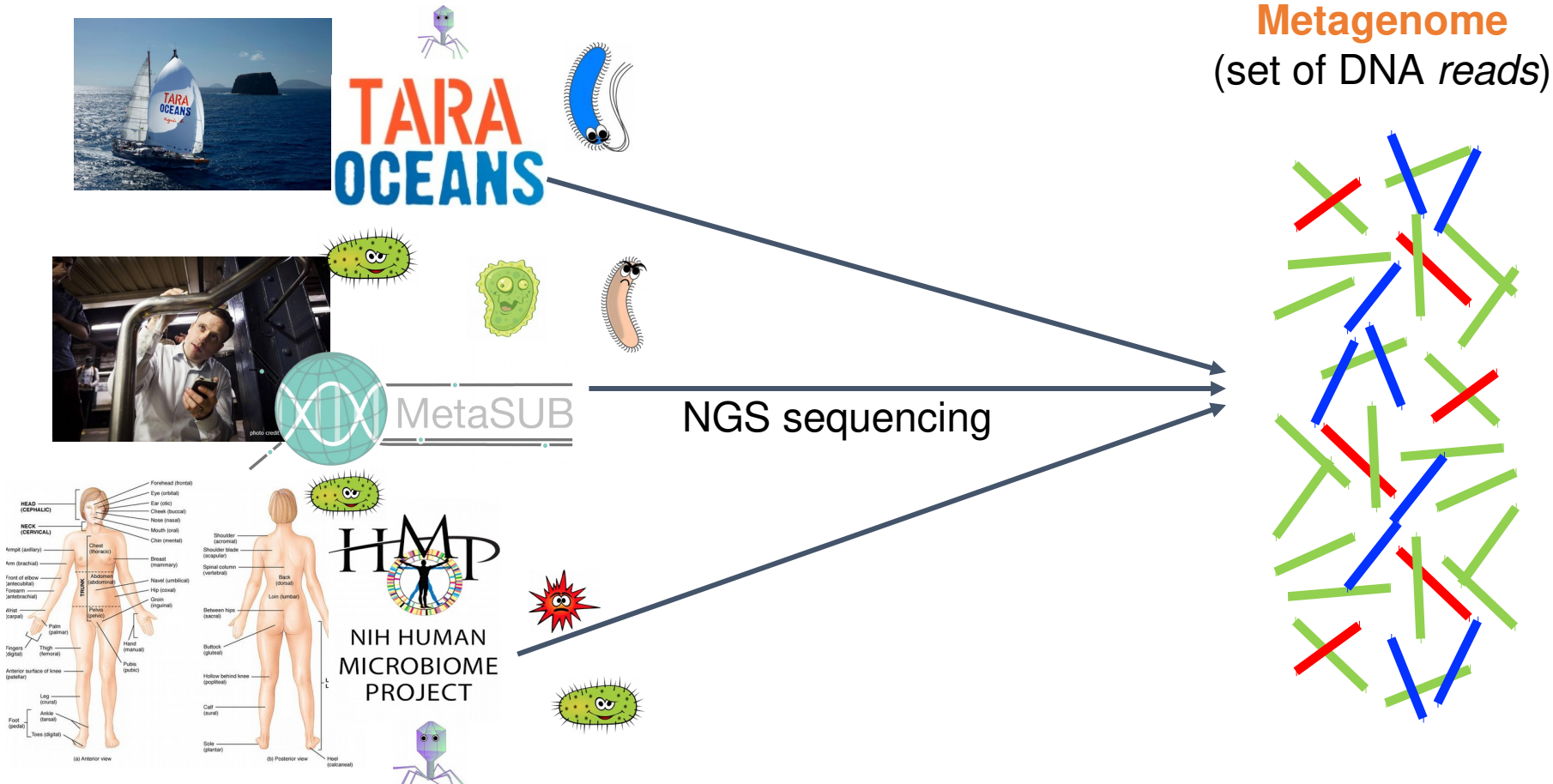
Hosted at HMS by:
Michael Baym
Karel Břinda

HARVARD MEDICAL SCHOOL

Overview

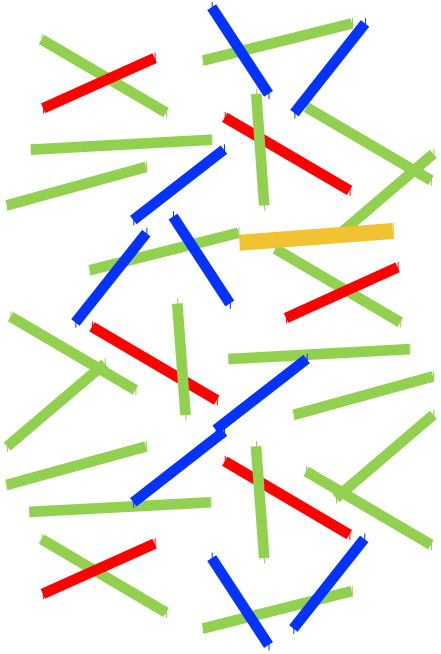
- What is Metagenomics
- Sequence classification with ProPhyle
- Estimation of abundances
- Experiments and validation
- Conclusions and perspectives

Metagenomic sequencing

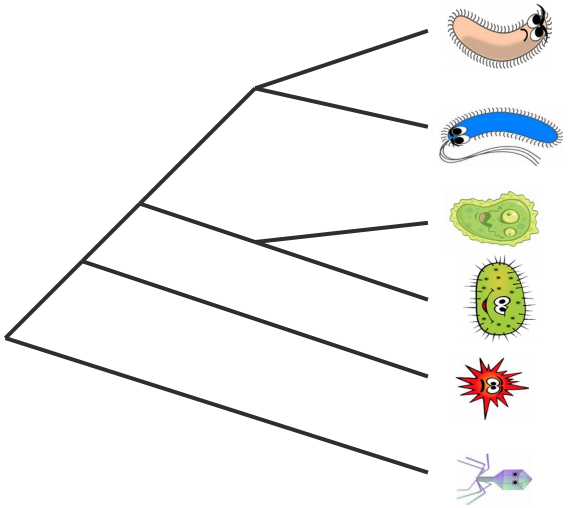


Phylogeny-based metagenomic classification

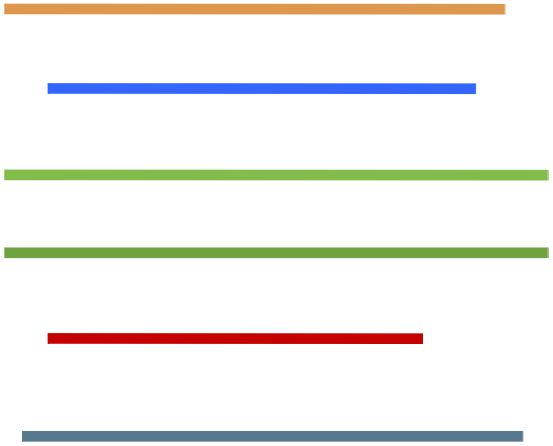
Metagenome reads
(billions)



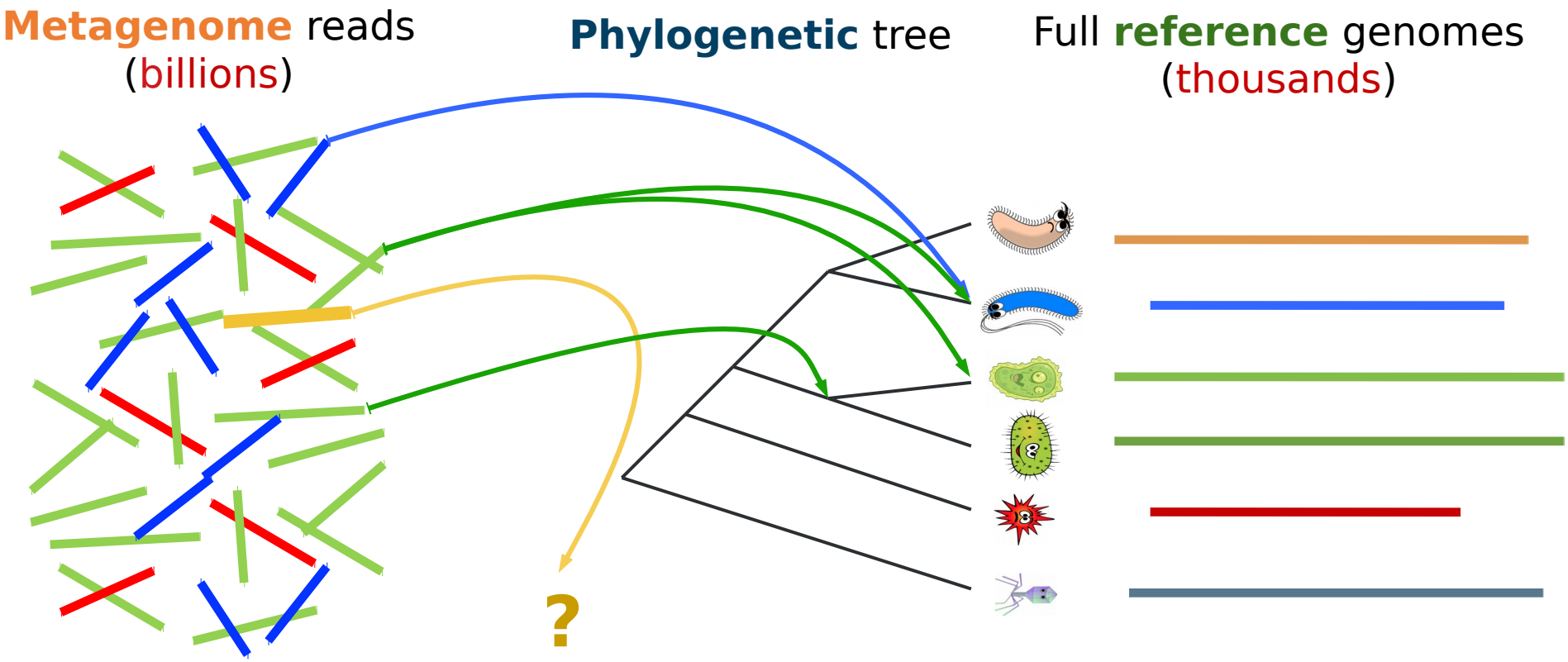
Phylogenetic tree



Full **reference** genomes
(thousands)



Phylogeny-based metagenomic classification

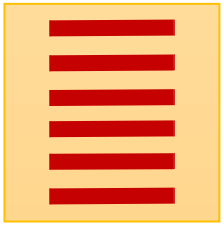


Alignment-free methods

Reference genome:

Index of *k*-mers

"Is this *k*-mer in the genome?" **yes**/**no**



Goal: estimate the "likelihood" of a read to belong to the genome (assign a score, e.g. number of hits)

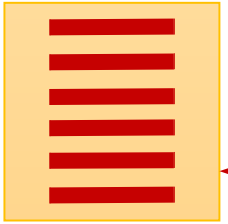
Read:

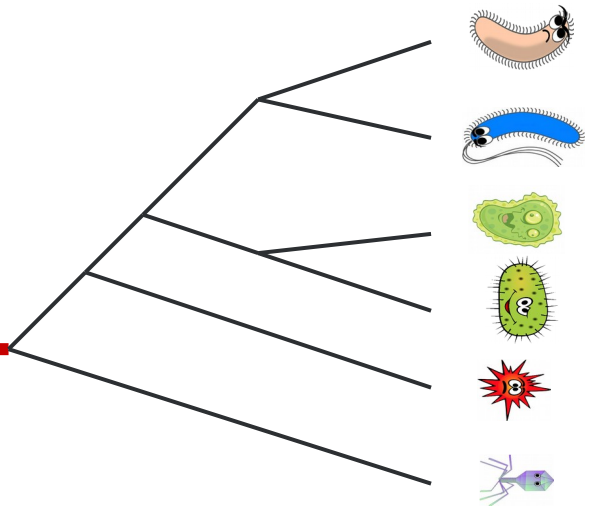
k-mers

Alignment-free methods

Index of *k*-mers

- Huge
- Repeated *k*-mers





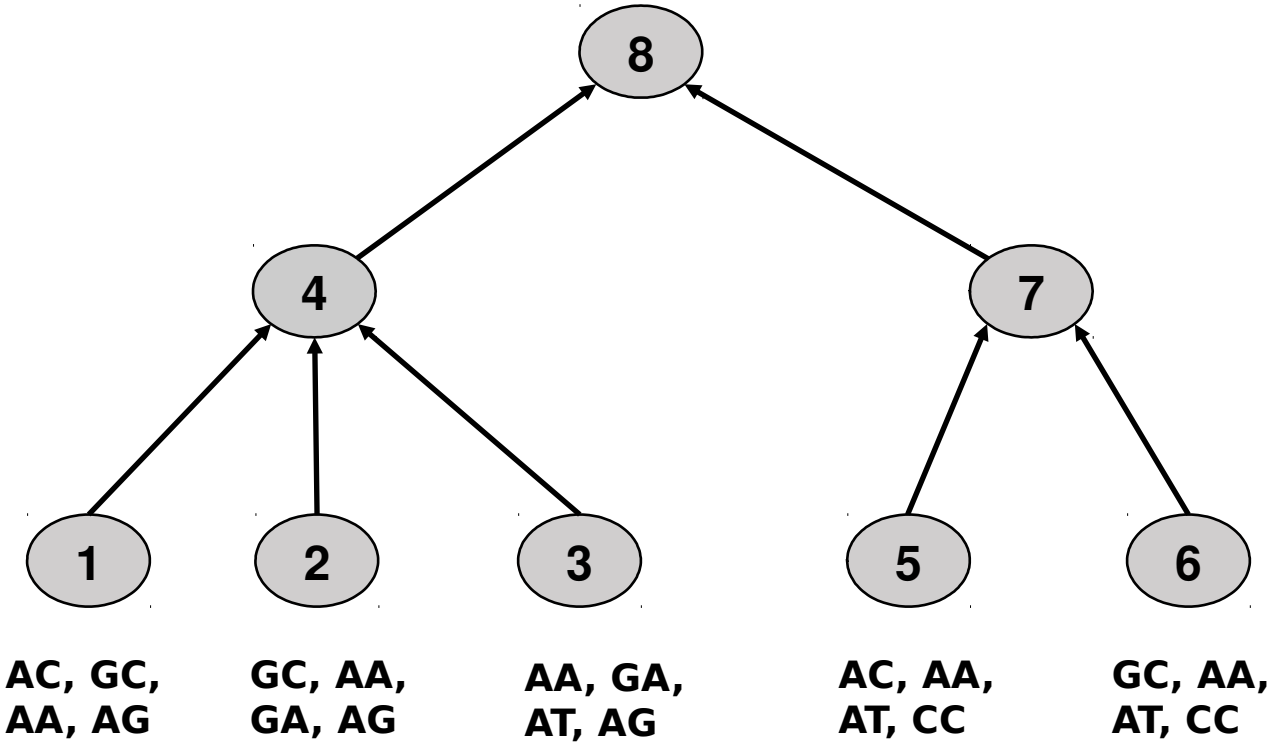
Read:

k-mers

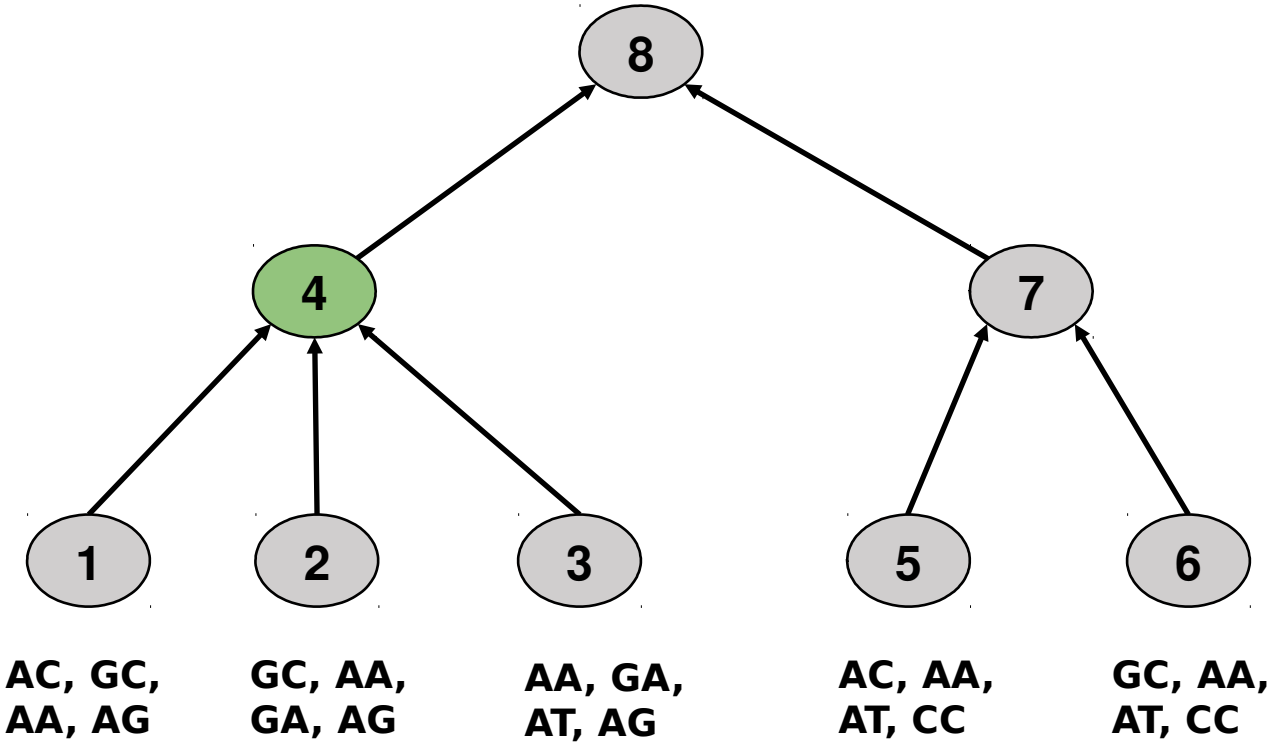
Same problem, with tree

How to **compress** it?

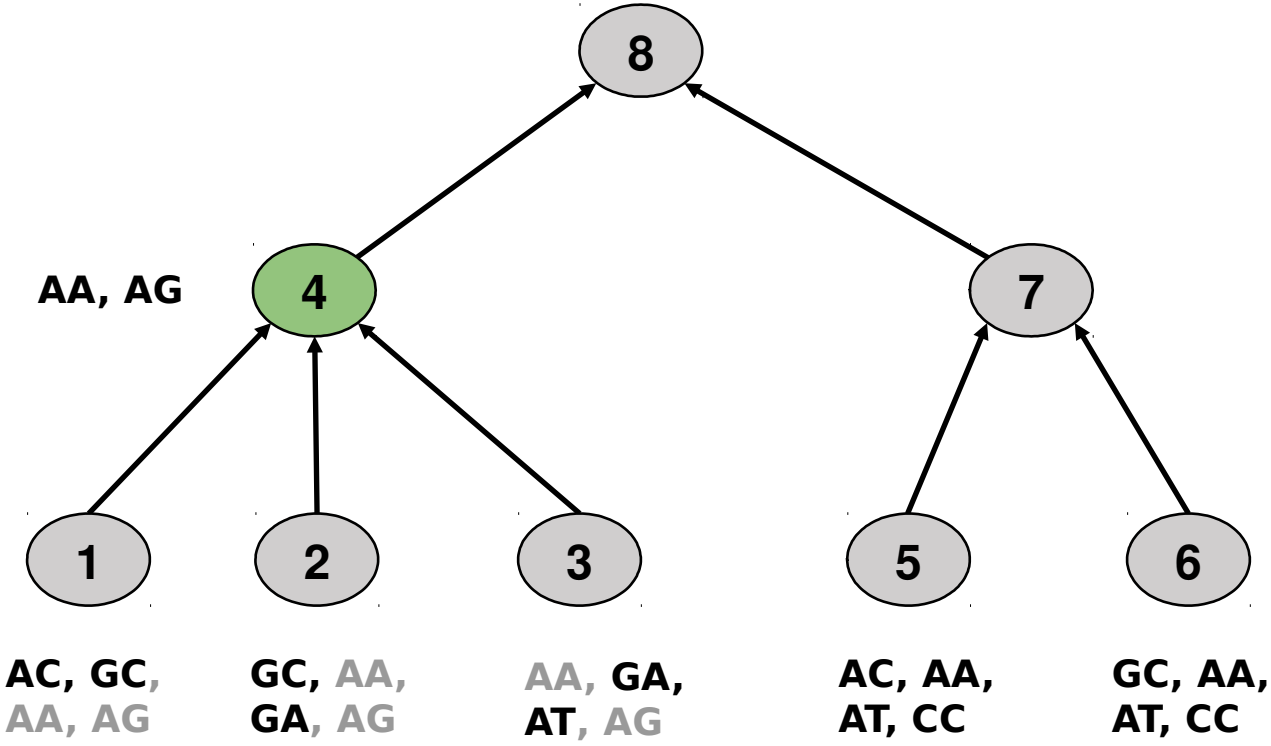
k-mer propagation - ProPhyle



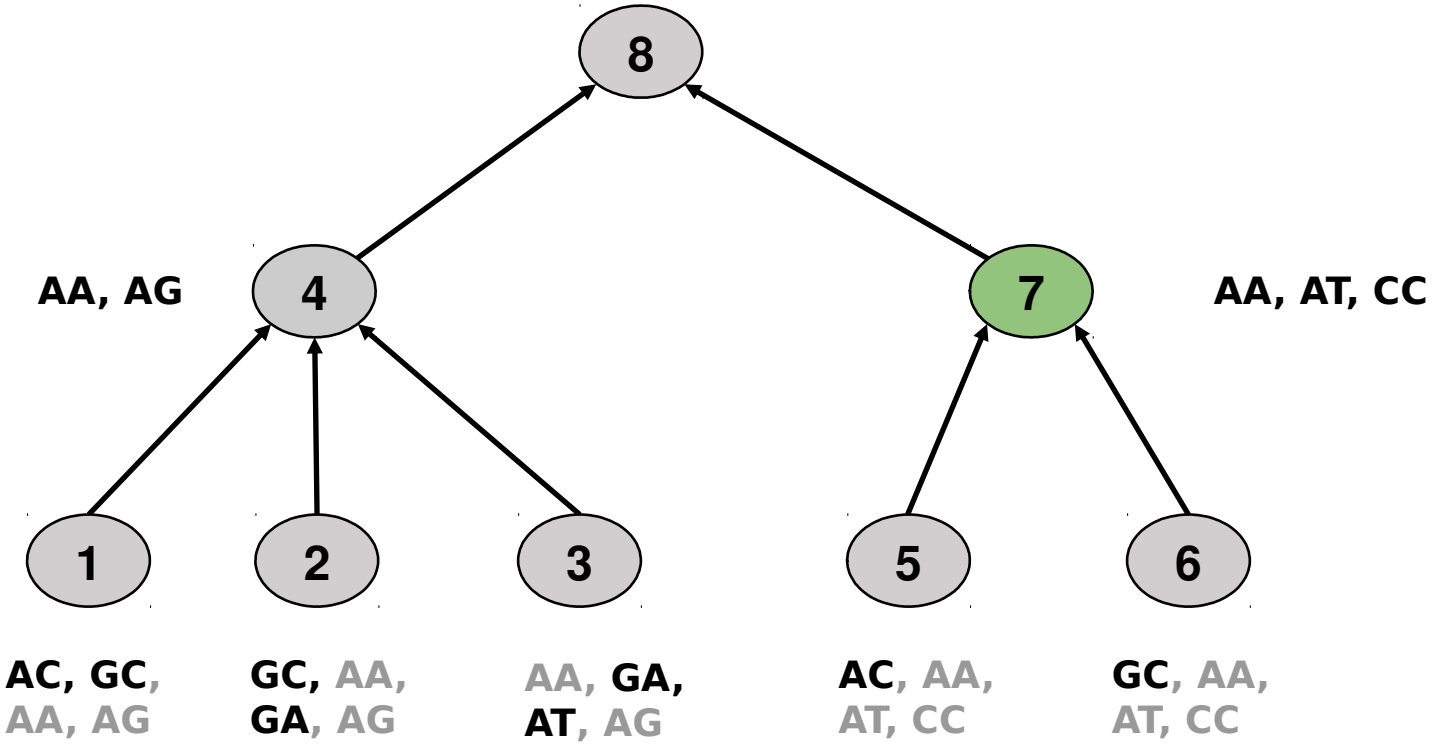
k-mer propagation - ProPhyle



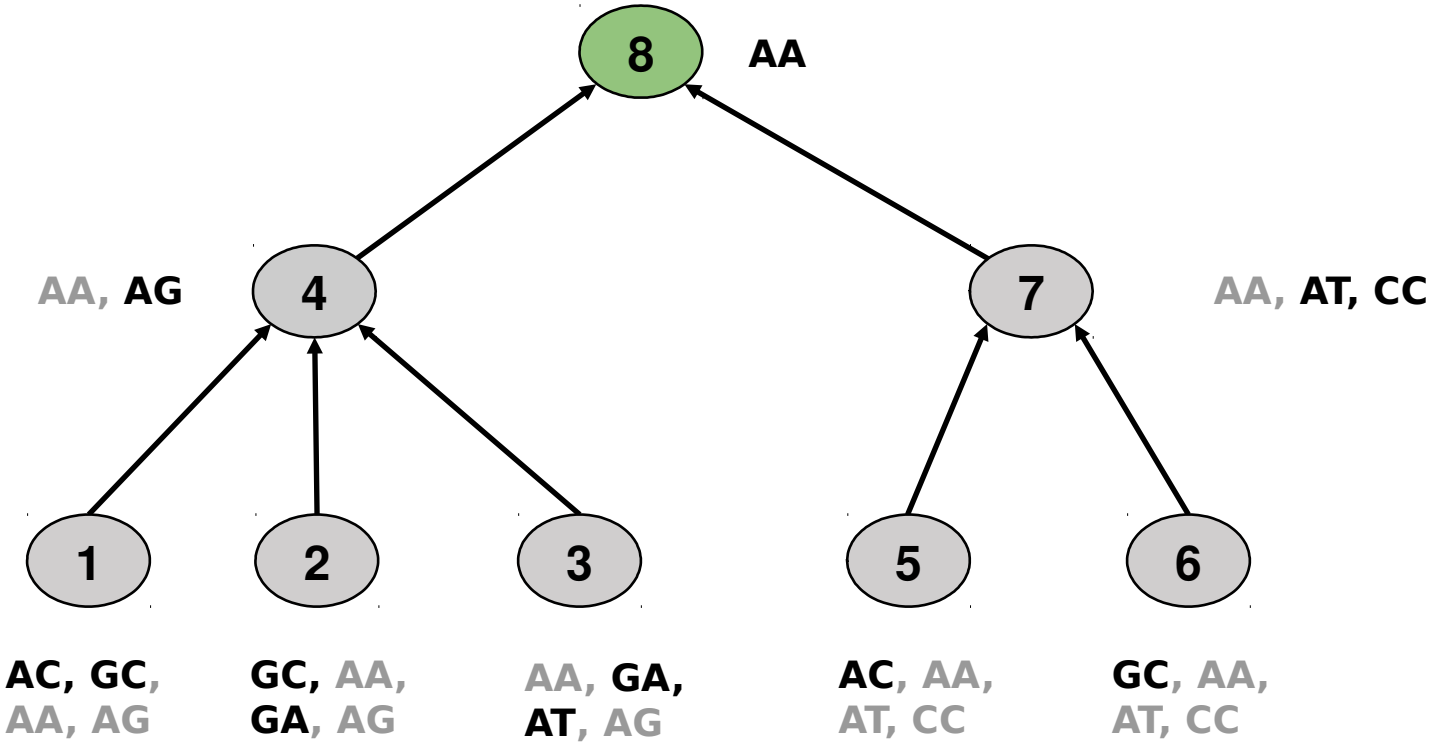
k-mer propagation - ProPhyle



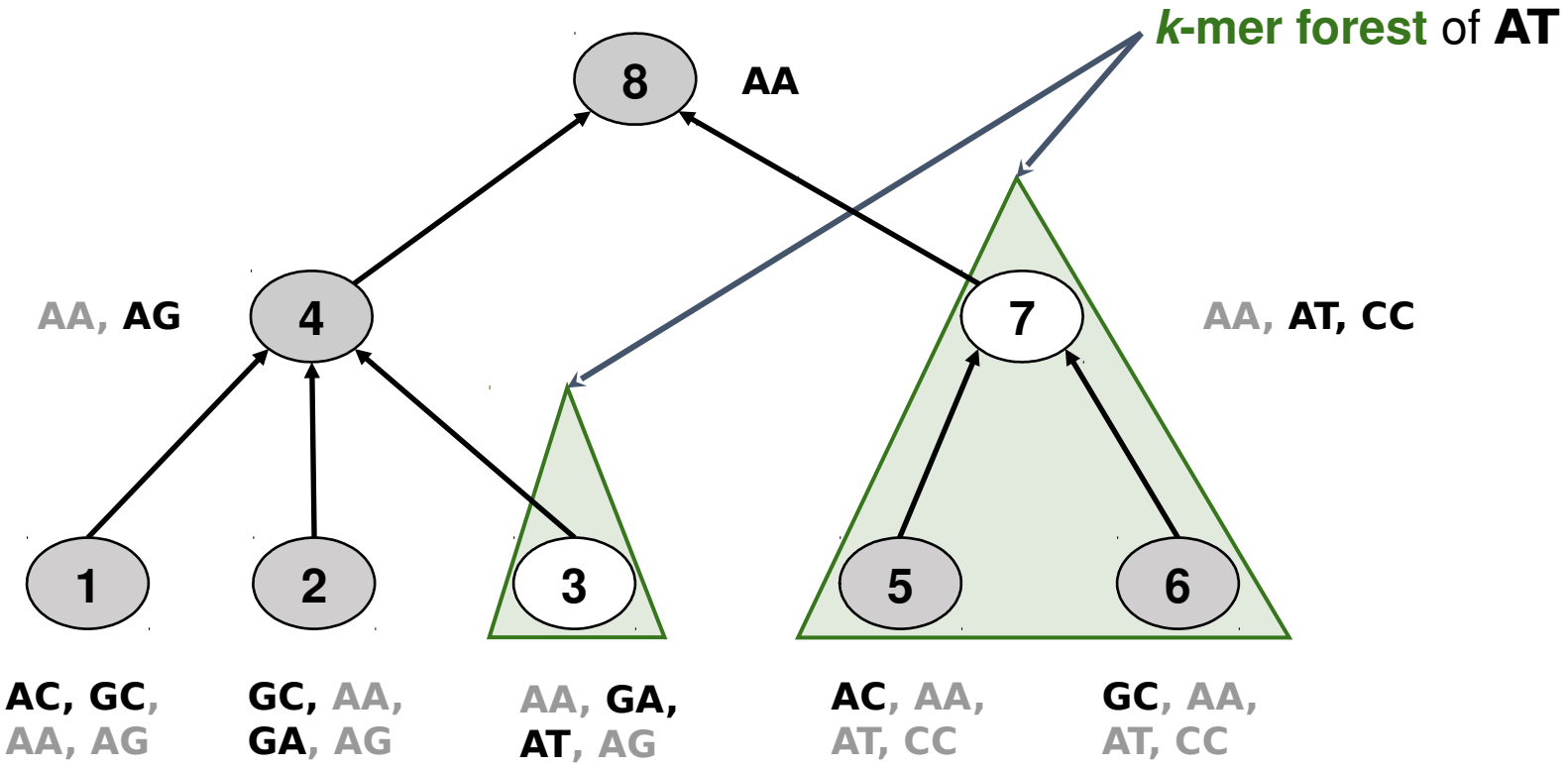
k-mer propagation - ProPhyle



k-mer propagation - ProPhyle

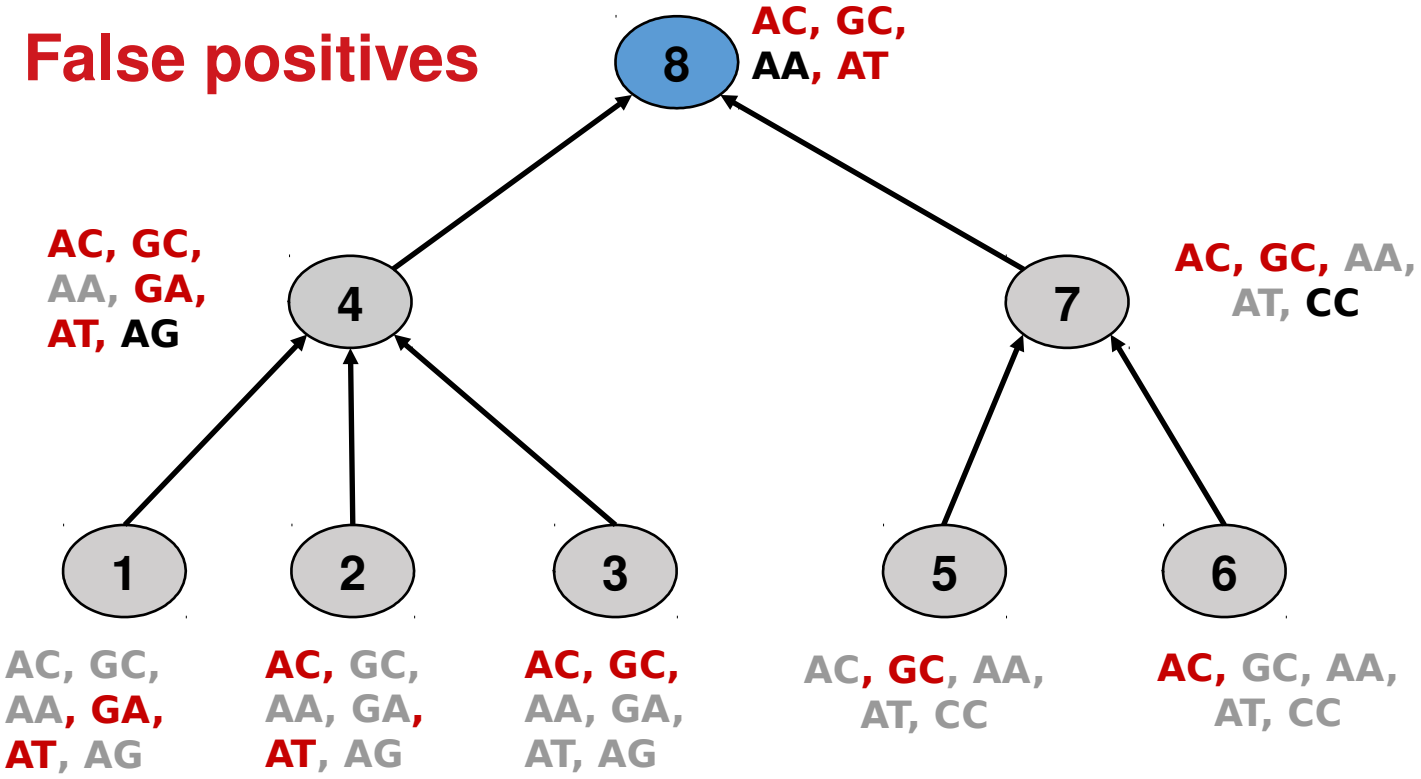


k-mer propagation - ProPhyle

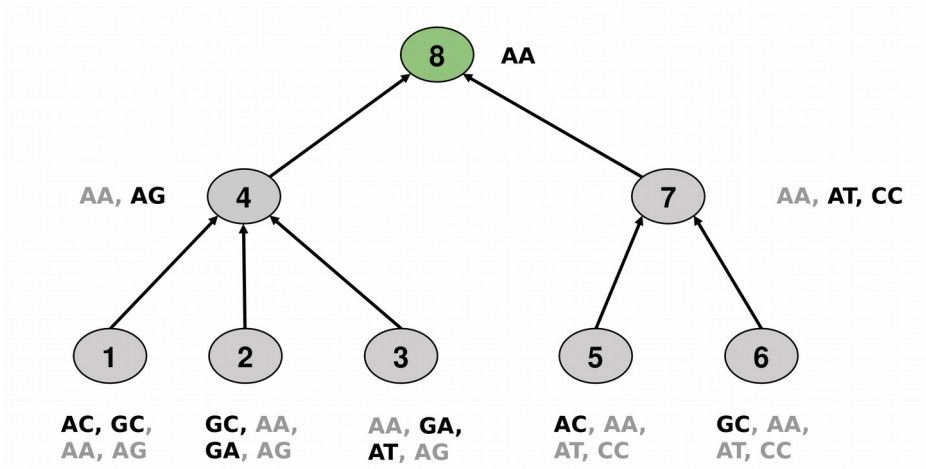


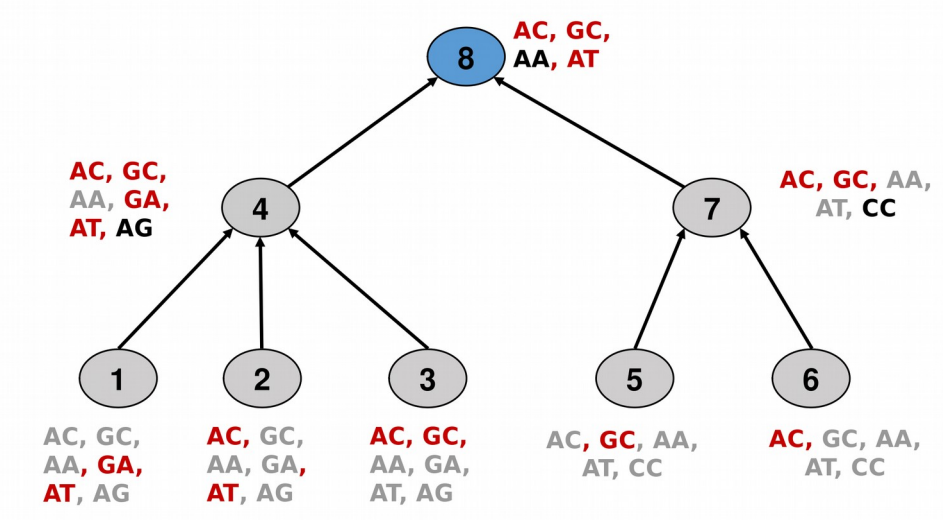
Kraken's LCA

(lowest common ancestors)



Assignments – **ProPhyle** vs **Kraken**

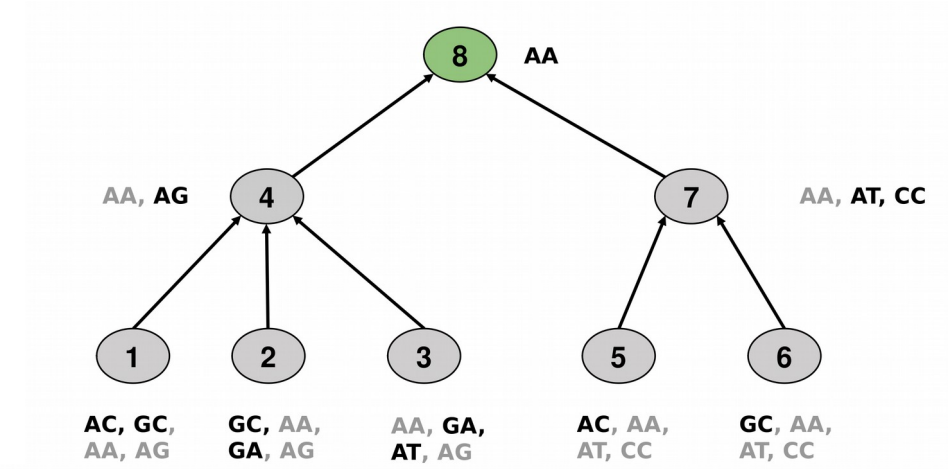


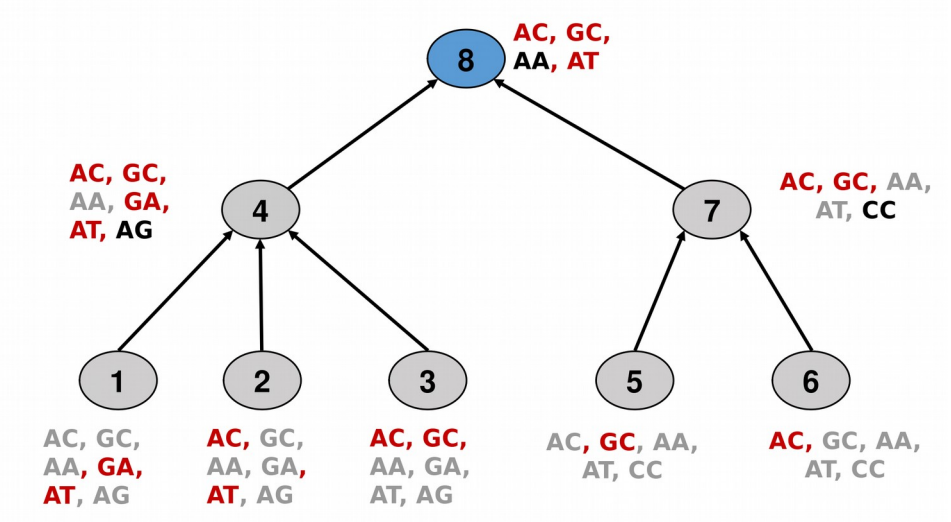


Read: AGC

k-mers: AG
GC

Assignments – ProPhyle vs Kraken



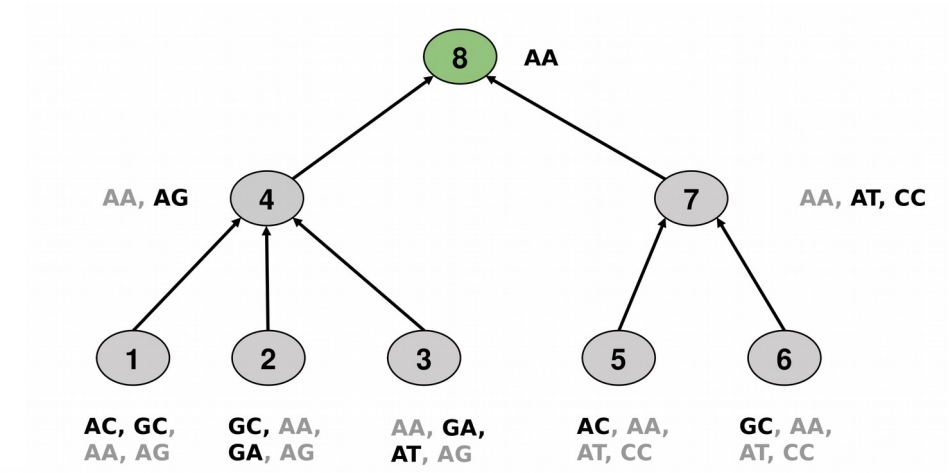


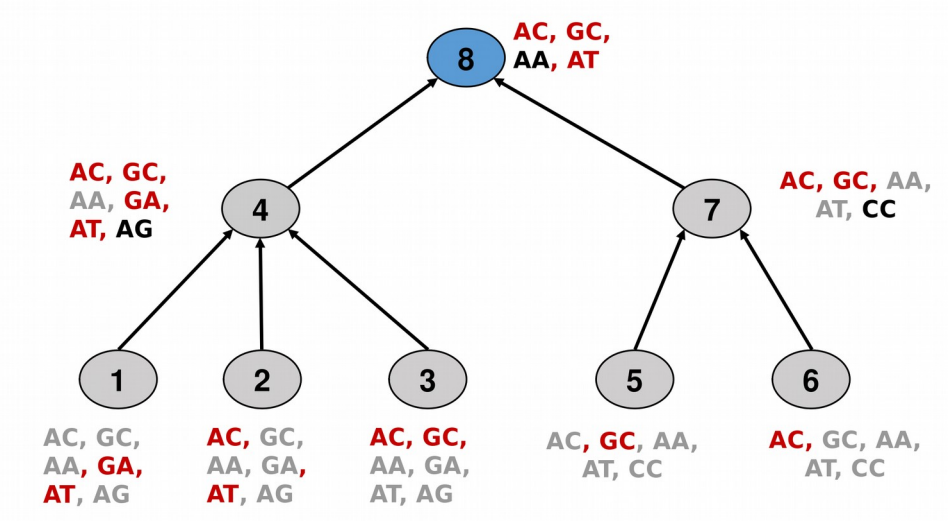
Read: AGC

k-mers: AG
GC

ProPhyle: AG → {4} → {1,2,3}
GC → {1,2}

Assignments – ProPhyle vs Kraken





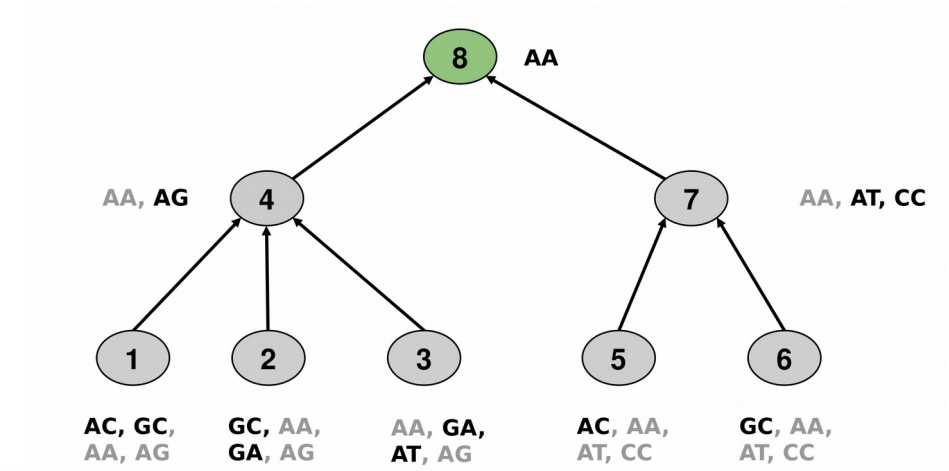
Read: AGC

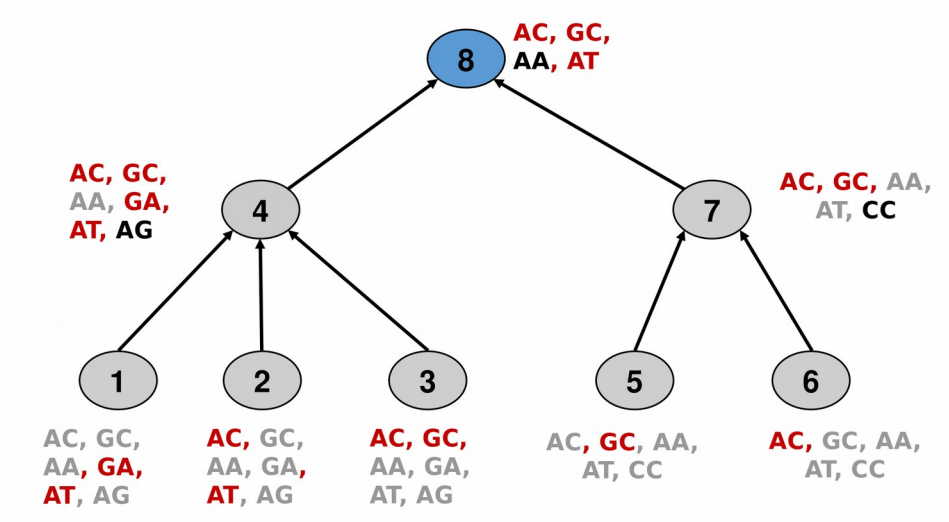
k-mers: AG
GC

ProPhyle: AG → {4} → {1,2,3}
GC → {1,2,6}
s(1) = 2
s(2) = 2
s(3) = 1
s(6) = 1
s(*) = 0

read → {1,2}

Assignments – ProPhyle vs Kraken



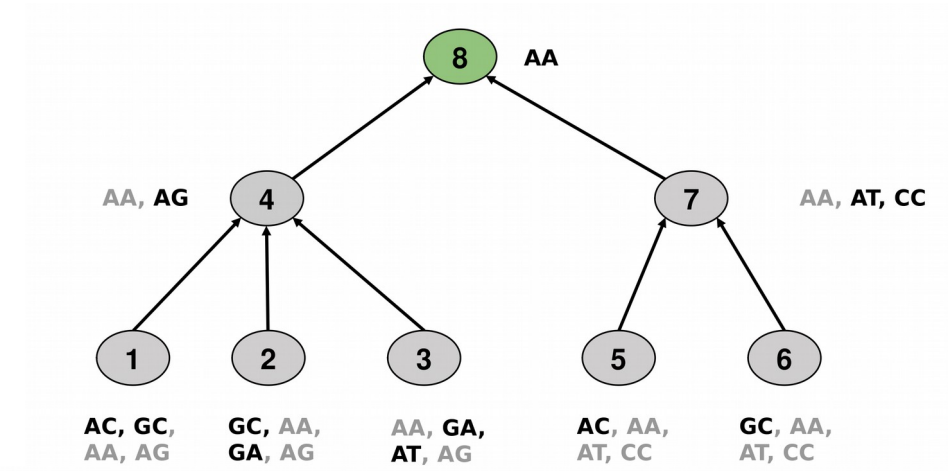


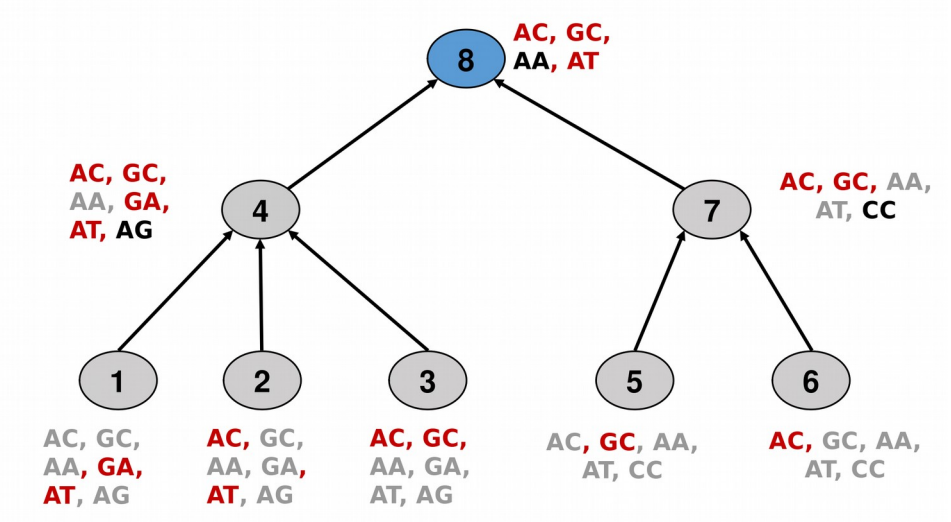
Read: AGC

k-mers: AG
GC

Kraken: AG → 4 → {1,2,3}
GC → 8 → {*}

Assignments – ProPhyle vs Kraken





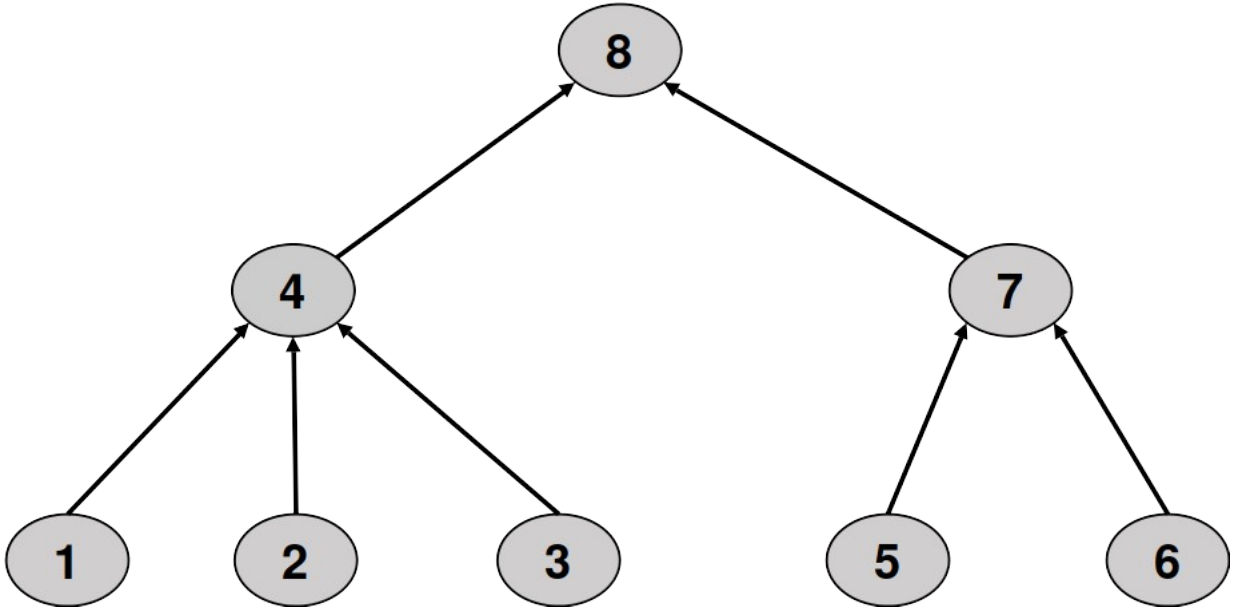
Read: AGC

k-mers: AG
GC

Kraken:

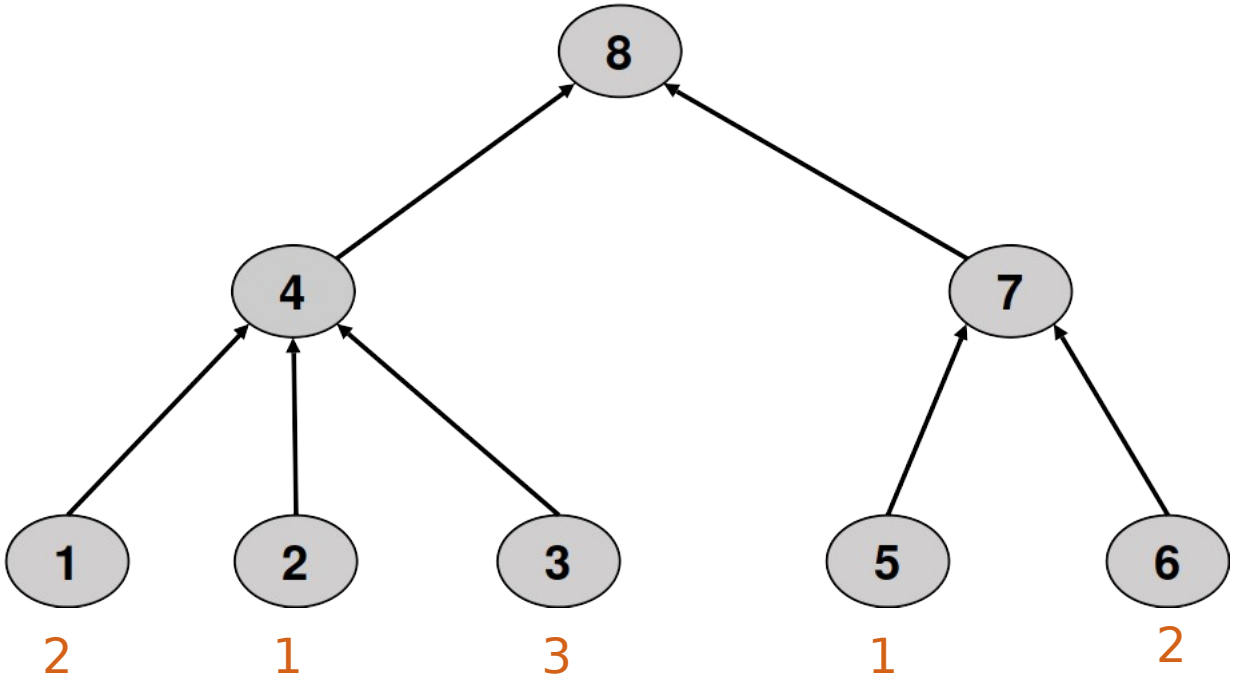
```
AG → 4 → {1,2,3}
GC → 8 → {*}
s(1) = 2
s(2) = 2
s(3) = 2
S(*) = 1

read → 4
4 = LCA({1,2,3})
```

What to do with multiple assignments?




```
R1 → {1,2}
R2 → {3,7} → {3,5,6}
R3 → {3}
R4 → {1,3,6}
```

What to do with multiple assignments?




```
R1 → {1,2}
R2 → {3,7} → {3,5,6}
R3 → {3}
R4 → {1,3,6}
```

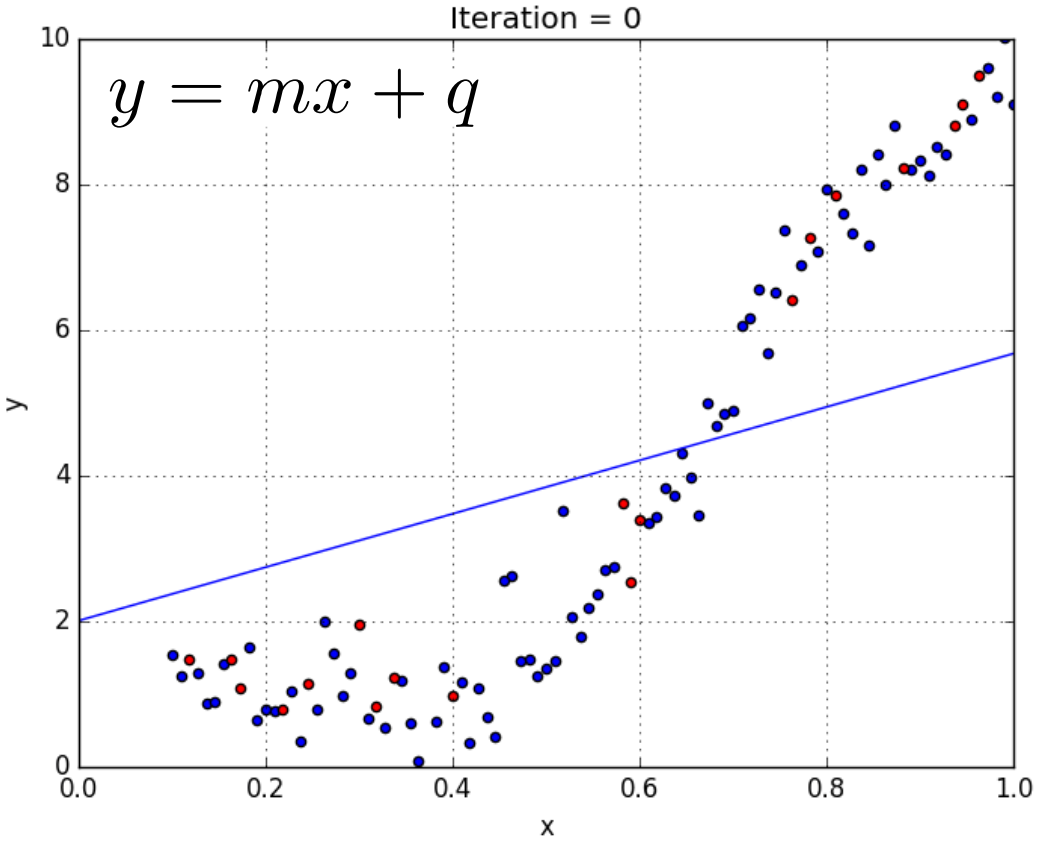
"Redistribute" reads

Consider all $\mathbf{2^n - 1}$ possible combinations

$\sim\mathbf{10^{301}}$ for $n = 1000$

$\Sigma = 9$ but only 4 reads...

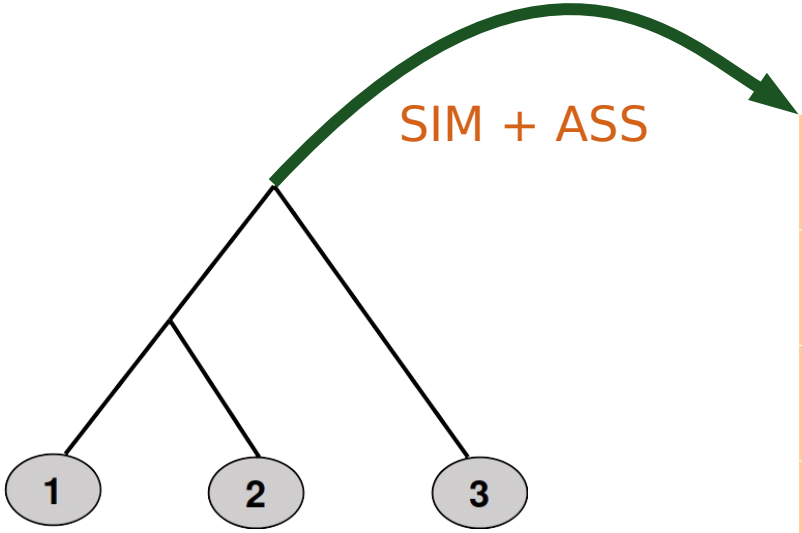
Linear Regression!



Acquiring statistics about mult. ass.

G_3:

```
R1        → {1,2,3}
R2        → {2,3}
R3        → {3}
R[4-10]   → {3} (x6)
```



S	G_1	G_2	G_3
G_1	**1**	0.1	**0.1**
G_2	0.2	**1**	0.2
G_3	0	0.1	**1**

"10% of reads simulated from genome 3 were assigned to a set containing G_1"

“Redistributing” multiple assignments

$$m = S \cdot r \quad \Rightarrow \quad \begin{cases} m_1 = r_1 + s_{12} r_2 + s_{13} r_3 \\ m_2 = s_{21} r_1 + r_2 + s_{23} r_3 \\ m_3 = s_{31} r_1 + s_{32} r_2 + r_3 \end{cases}$$

S	G_1	G_2	G_3
G_1	**1**	0.1	**0.1**
G_2	0.2	**1**	0.2
G_3	0	0.1	**1**

$\bullet$

r
?
?
?

=

m
11
4
10

$$\mathrm{r} = \arg\min_r \|m - S \cdot r\|^2$$

“Redistributing” multiple assignments

$$m = S \cdot r \quad \Rightarrow \quad \begin{cases} m_1 = r_1 + s_{12} r_2 + s_{13} r_3 \\ m_2 = s_{21} r_1 + r_2 + s_{23} r_3 \\ m_3 = s_{31} r_1 + s_{32} r_2 + r_3 \end{cases}$$

S	G_1	G_2	G_3
G_1	**1**	0.1	**0.1**
G_2	0.2	**1**	0.2
G_3	0	0.1	**1**

$\bullet$

r
10
0
10

=

m
11
4
10

Same thing in
[**#ref.gen.**] dimensions

Need regularization

- Many assignments are inaccurate
  - sequencing errors
  - something not in DB
  - *k*-mers only heuristic

- 10^9 reads + 10^5 ref. Genomes → anything could get few assignments

- Real scenario: 100s of known organisms in a sample → 1/100 of index

- Approximate the system while keeping results sparse

- Introduce penalties for “using too many variables”

LASSO regressor

$$\mathrm{r} = \arg\min_r(\|m - S \cdot r\|^2 + \lambda_1 \|r\|_1)$$

Elastic Net
(Zou & Hastie, 2005)

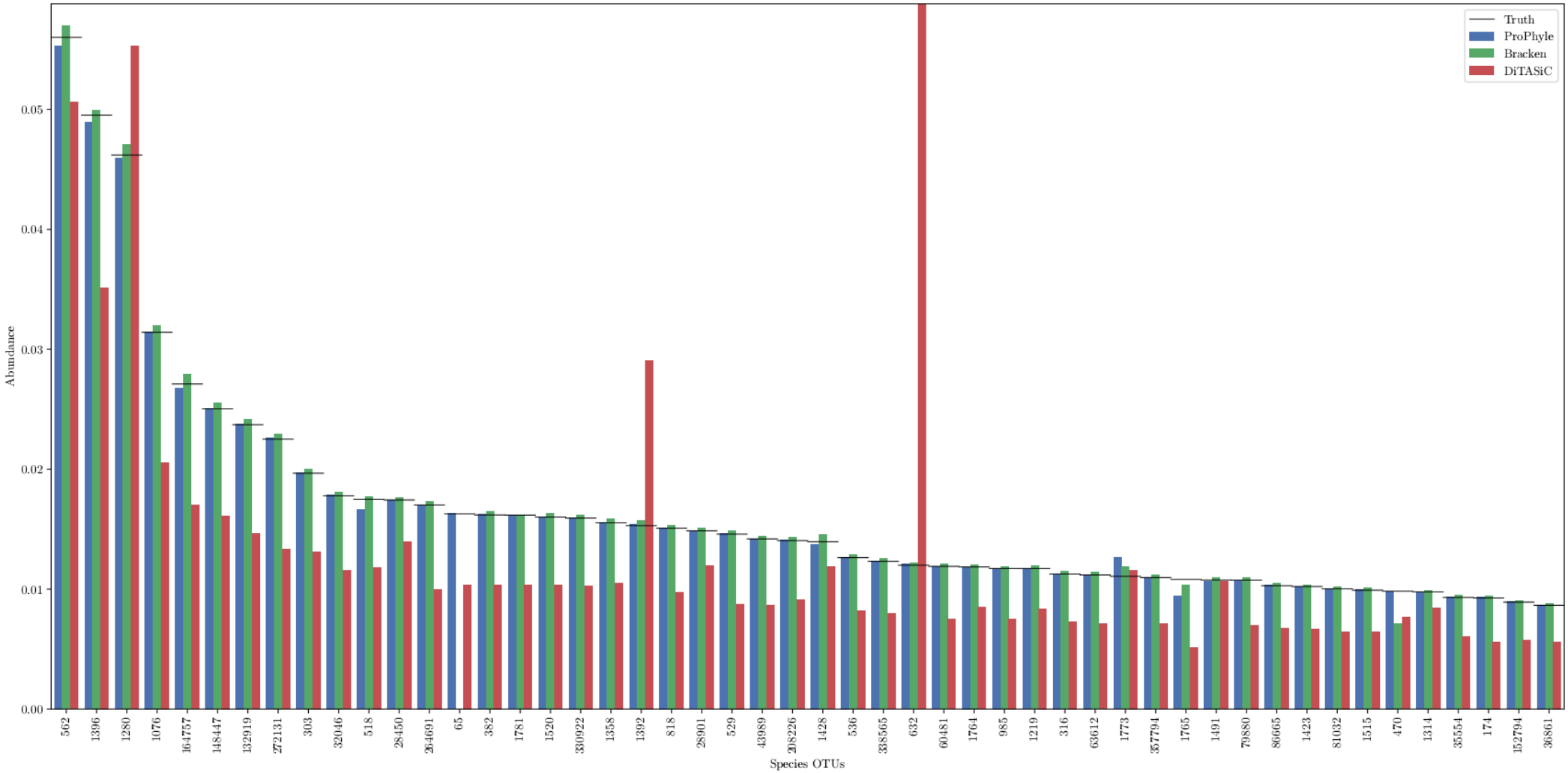
- L_2 regularization does not set regr. coeff. to 0
- Lasso (L_1) → among highly correlated variables will choose only 1 (at **random**)
-  Combine L_1 and L_2!
- The quadratic part of the penalty:
  - encourages ***grouping*** effect
  - ***stabilize*** regularization path (selected variables)

$$\mathrm{r} = \arg\min_r(\|m - S \cdot r\|^2 + \lambda_1\|r\|_1 + \lambda_2\|r\|_2^2)$$

Can also simulate LASSO, which has desirable effects

Simulation experiment (Mende et al, 2012)

- ~25M 75bp reads simulated from 193 genomes (85 species)
- includes multiple strains in the genera Bacillus and Mycobacterium
  - notoriously difficult to estimate ab. for (due to high similarity)
- Index for **ProPhyle**, **Bracken** and **DiTASiC** containing 1267 genomes
  - including simulated ones
  - $k = 31 \rightarrow 45$ k-mers per read
- Residual Sum of Squares (RSS) error measure: $$\sum_{i=1}^{n} \varepsilon^2 = \sum_{i=1}^{n} (t_i - e_i)^2$$


Truth
ProPhyle
Bracken
DiTASiC
Abundance
0.00
0.01
0.02
0.03
0.04
0.05
562
1396
1280
1076
164757
148447
132919
272131
303
32046
518
28450
264691
65
382
1781
1520
330922
1358
1392
818
28901
529
43989
208226
1428
536
338565
632
60481
1764
985
1219
316
63612
1773
357794
1765
1491
79880
86665
1423
81032
1515
470
1314
35554
174
152794
36861
Species OTUs

Results - Mende

		LASSO	E-Net	Bracken	DiTASiC
Genome	RSS	**7.52e-4**	9.32e-4	NA	2.26e-02
	FP# (ab)	**32 (5.60e-02)**	89 (8.57e-2)	NA	648 (3.22e-01)
	FN# (ab)	66 (1.06e-02)	**2 (2.02e-4)**	NA	32 (2.07e-02)
Species	RSS	3.84e-05	**6.68e-6**	2.80e-04	4.53e-02
	FP# (ab)	**1 (1.44e-04)**	26 (1.36e-3)	170 (9.73e-04)	381 (6.08e-02)
	FN# (ab)	**0 (0)**	0 (0)	1 (1.63e-02)	0 (0)
Genus	RSS	1.33e-05	**1.16e-06**	2.85e-04	5.45e-02
	FP# (ab)	**0 (0)**	12 (3.62e-5)	78 (3.23e-04)	194 (1.06e-02)
	FN# (ab)	**0 (0)**	0 (0)	1 (1.63e-02)	0 (0)

“Stress Test”

- 10 samples with 500k read pairs of length 150bp
- Simulated from ref (250 genomes each, exponential distribution)
- Rare genomes (~50) have 10^{-5} abundance → 5 reads
- Index for all (6171) “reference” and “representative” RefSeq genomes (Archaea, Bacteria, Fungi, Viruses)
- ProPhyle index: ~50GB **vs** Kraken DB: ~330GB
- Kraken running for weeks on a powerful cluster

Results – Stress

- OK scalability (~5 days to generate simulation matrix with 0.3 cov)
- As for Mende, can effectively use LASSO for FP → 0
- Grid-Search for optimal parameters in few minutes (needs ground truth)
- Many FN as intended (genomes with <10 reads)

ISSUES:

- Parameters choice extremely important
- Most samples fit perfectly (Pearson >0.95), 2 of them **don't** (<0.4)
- Need more statistical info (e.g. *P*-values or other confidence est.)
- Cannot scale to index with ~100k ref

Real – HMP pilot

- 6.5M reads of length 75bp
- Even mixture of DNA from 22 isolates
- Same index as *stress* (6171 RefSeq genomes)
- 6 out of 22 isolates only have relatives in the index
- 1 fungus with 18S gene only (assignments suggest low copy number)

Results - HMP

		LASSO
Genome	RSS	0.15
	FP# (ab)	5 (0.01)
	FN# (ab)	3 (0.14)
Species	RSS	0.15
	FP# (ab)	3 (6.30e-03)
	FN# (ab)	1 (0.05)
Genus	RSS	0.15
	FP# (ab)	1 (5.66e-03)
	FN# (ab)	0 (0)

Results - HMP

		LASSO
Genome	RSS	0.15
	FP	5 (0.01)
	FN	3 (0.14)
Species	RSS	0.15
	FP	3 (6.30e-03)
	FN	1 (0.05)
Genus	RSS	0.15
	FP	1 (5.66e-03)
	FN	0 (0)

Illumina sequencing of HMP Mock Community even sample (SRR172902)

Metadata | Analysis (alpha) | Reads | Download

Warning: experimental software

Taxonomy Analysis

Unidentified reads: 28.13%

Identified reads: 71.87%

- cellular organisms: **71.87%**
  - Bacteria: **71.37%**
    - Terrabacteria group: **41.44%**
      - Deinococcus-Thermus: **23.05%**
        - Deinococcus: **23.04%**
          - Deinococcus radiodurar
            - Deinococcus radiodur
          - Deinococcus wulumuqie
      - Firmicutes: **13.51%**
        - Bacilli: **10.84%**
          - Bacillales: **5.3%**
          - Lactobacillales: **4.49%**
        - Clostridia: **2.59%**
      - Actinobacteria: **4.3%**
    - Proteobacteria: **17.69%**
      - Gammaproteobacteria: **10.**
      - delta/epsilon subdivisions: **2**
      - Alphaproteobacteria: **2.07%**
      - Betaproteobacteria: **1.78%**
    - FCB group: **6.82%**
      - Bacteroidales: **6.81%**
        - Bacteroides: **6.4%**
          - Bacteroides vulgatus: **1**
  - Archaea: **0.5%**
  - Eukaryota: **0.01%**

Strong signals

SuperKingdom	Organism	Rank	%%	Kbp	weighted score
Bacteria	Deinococcus radiodurans	species	25.3	124,506	**38.2**
Bacteria	Acinetobacter baumannii	species	11.3	55,472	**13.9**
Bacteria	Bacteroides vulgatus	species	7.4	36,299	**7.1**
Bacteria	Propionibacterium	genus	3.6	17,558	17.6
Bacteria	Staphylococcus aureus	species	3.5	17,334	**6.1**
Bacteria	Streptococcus mutans	species	2.9	14,255	**7.3**
Bacteria	Clostridium beijerinckii	species	2.9	14,086	**2.4**
Bacteria	Helicobacter pylori	species	2.6	13,027	**8.0**
Bacteria	Rhodobacter sphaeroides	species	2.4	11,687	**2.5**
Bacteria	Streptococcus pneumoniae	species	2.1	10,342	**5.0**
Bacteria	Neisseria meningitidis	species	2.0	10,052	**4.7**
Bacteria	Listeria monocytogenes	species	1.8	9,057	**3.0**
Bacteria	Actinomyces odontolyticus ATCC 17982		1.1	5,646	5.6
Bacteria	Bacillus cereus group	species group	0.7	3,404	3.4
Bacteria	Pseudomonas	genus	0.6	3,158	3.2
Archaea	Methanobrevibacter smithii	species	0.5	2,472	**1.3**
Bacteria	Enterococcus	genus	0.4	2,015	2.0

Adjusted Results - HMP

		LASSO
Genome	RSS	**0.03**
	FP	5 (0.01)
	FN	3 (**2.28e-6**)
Species	RSS	**0.03**
	FP	3 (6.30e-03)
	FN	1 (**7.61e-07**)
Genus	RSS	**0.03**
	FP	1 (5.66e-03)
	FN	0 (0)

Suggests problem with:

- reference genomes (e.g. contamination)
- sample preparation (e.g. not so even mix)
- current computational approaches

Conclusions and perspectives

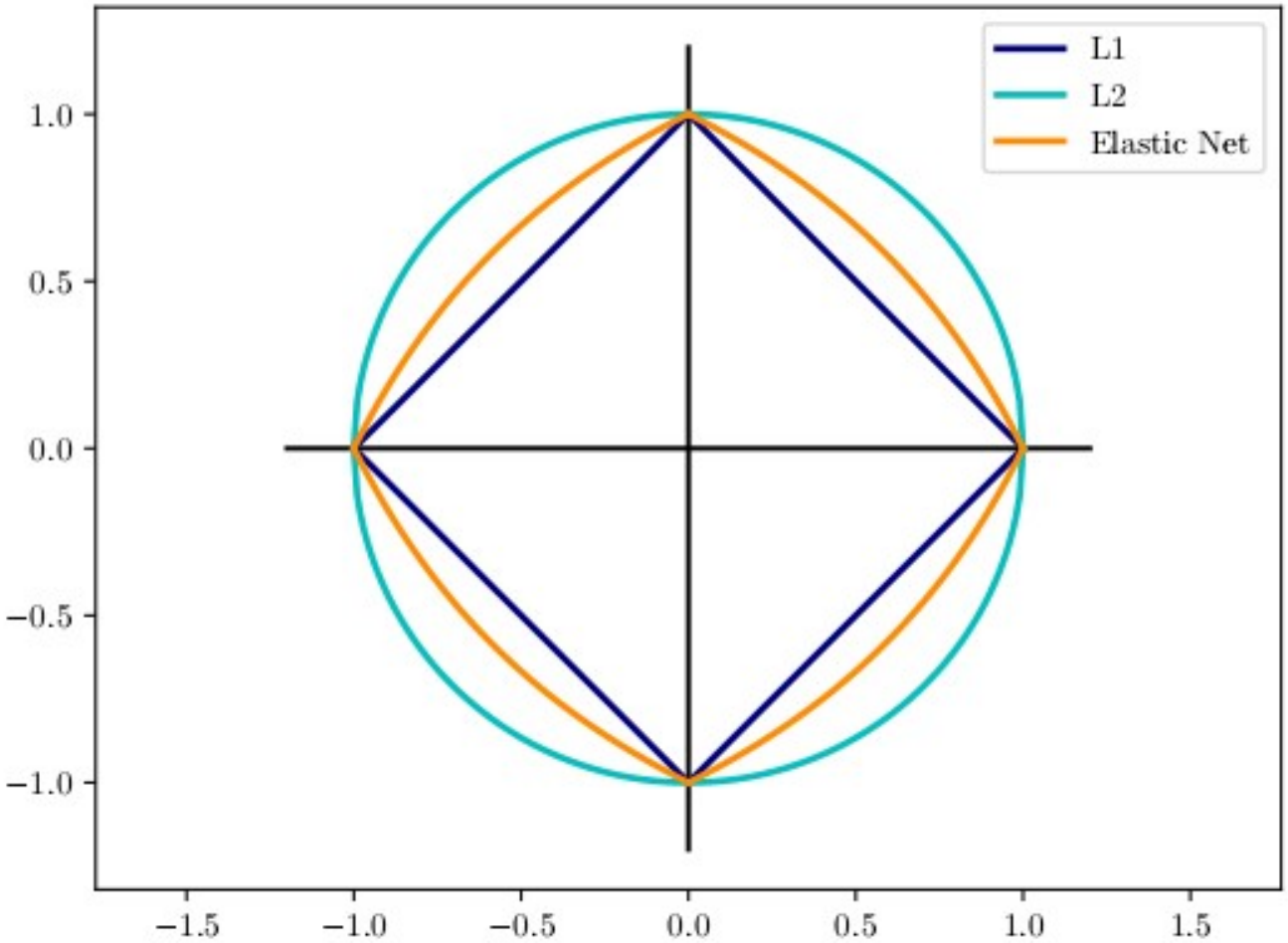
- **ProPhyle** is a complete, **resource-frugal** and easy-to-use metagenomic classifier
- Lossless index, suitable for inaccurate phylogenetic trees
- Flexibility and feature richness (works with any tree, standard bioinformatics formats)

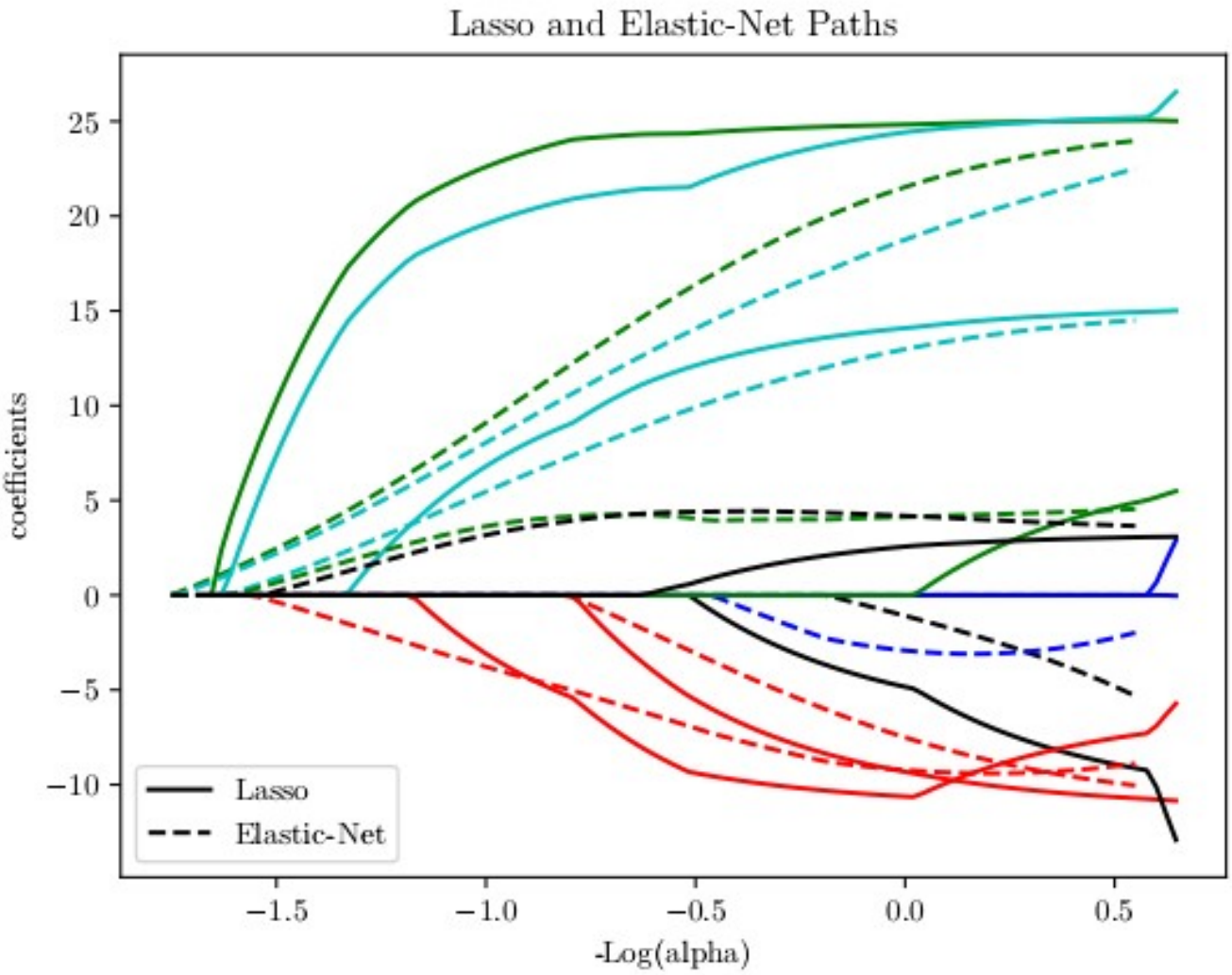
Future directions:

- Assignment quality and read/ref.genome length heuristics for abundances
- ML framework to estimate optimal reg. parameters based on:
  - complexity of sample
  - # ref. genomes
  - sequencing technology

github SOCIAL CODING	http://github.com/karel-brinda/prophyle
Read the Docs	http://prophyle.rtfd.io
BIOCONDA	$ conda install prophyle
python Package Index	$ pip install prophyle


Thank you!
© Scott Chimileski


L1
L2
Elastic Net
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Lasso and Elastic-Net Paths
coefficients
25
20
15
10
5
0
−5
−10
−1.5
−1.0
−0.5
0.0
0.5
-Log(alpha)
Lasso
Elastic-Net