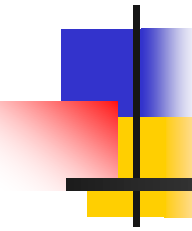
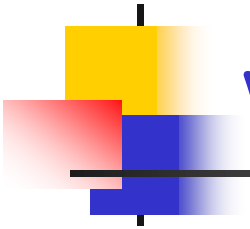


RandNLA:
Randomization in Numerical Linear Algebra

Petros Drineas

Department of Computer Science
Purdue University

For slides, etc. google drineas



Why RandNLA?

Randomization and sampling allow us to design provably accurate algorithms for problems that are:

- Massive

(matrices so large that can not be stored at all, or can only be stored in slow memory devices)

- Computationally expensive or NP-hard

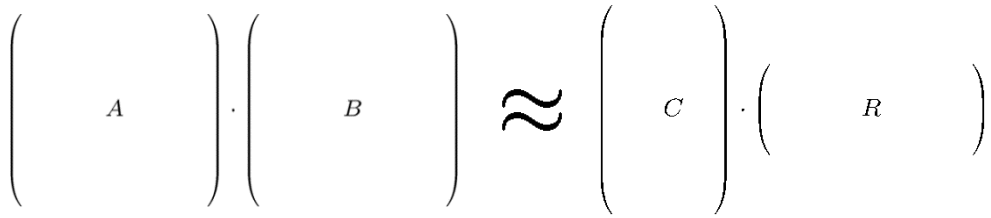
(combinatorial optimization problems, such as the Column Subset Selection Problem)

RandNLA in a slide

Randomized algorithms

• By (carefully) sampling rows/columns of a matrix, we can construct new, smaller matrices that are close to the original matrix (w.r.t. matrix norms) with high probability.

Example: Randomized Matrix Multiplication



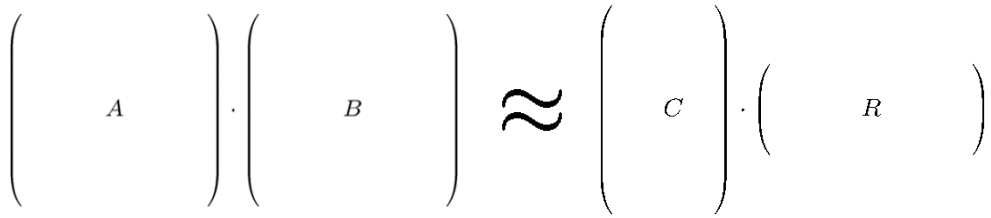
Here C consists of a few (rescaled) columns of A and R consists of the corresponding (rescaled) rows of B.

RandNLA in a slide

Randomized algorithms

• By preprocessing the matrix using almost any random matrix, we can sample rows/columns much less carefully (uniformly at random) and still get nice bounds with high probability.

Example: Randomized Matrix Multiplication



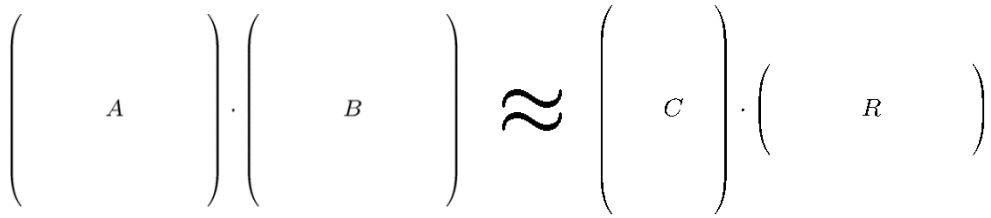
This is equivalent to setting $C = AX$, where X is (say) a random Gaussian matrix and the number of columns in X is much smaller than the number of columns in A; similarly, $R = X^T B$.

RandNLA in a slide

Randomized algorithms

• By (carefully) sampling rows/columns of a matrix, we can construct new, smaller matrices that are close to the original matrix (w.r.t. matrix norms) with high probability.

Example: Randomized Matrix Multiplication



• By preprocessing the matrix using "random projection" matrices, we can sample rows/columns much less carefully (uniformly at random) and still get nice bounds with high probability.

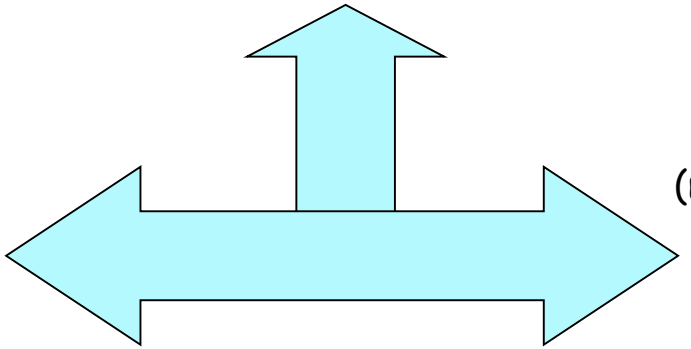
Matrix perturbation theory

• The resulting "sketches" behave similarly (e.g., in terms of singular values and singular vectors) to the original matrices thanks to the norm bounds.

Interplay

Applications in BIG DATA

(Data Mining, Information Retrieval, Machine Learning, Bioinformatics, etc.)



Theoretical Computer Science

Randomized and approximation algorithms

Applied Math

1. Numerical Linear Algebra
(matrix computations, perturbation theory)

2. Probability theory
(esp. measure concentration for sums of random matrices)

Roadmap

(Drineas & Mahoney, Lectures on RandNLA, Vol. 25, Amer. Math. Soc., 2018)

- **RandNLA approaches for regression problems**
- **RandNLA approaches for Principal Component Analysis (PCA)**

(was also discussed in Andreas Stathopoulos' talk on Monday)

Roadmap

(Drineas & Mahoney, Lectures on RandNLA, Vol. 25, Amer. Math. Soc., 2018)

- **RandNLA approaches for regression problems**
- **RandNLA approaches for Principal Component Analysis (PCA)**

(was also discussed in Andreas Stathopoulos' talk on Monday)

Why regression and PCA?

Both problems are of paramount importance in Big (as well as in Tiny, Small, Medium, Massive, etc.) Data analysis.

Both problems are at the heart of multiple disciplines: Computer Science (Numerical Linear Algebra, Machine Learning), Applied Mathematics, and Statistics.

Both problems have a very rich history:

- Regression was introduced in the 1800s (Gauss, Legendre, etc.)
- PCA was introduced in the 1900s (Pearson, Hotelling, etc.)

Problem definition and motivation

In data analysis applications one has n observations of the form:

$$y_i = y(t_i), i = 1, \ldots, n$$

Model y(t) (unknown) as a linear combination of d basis functions:

$$y(t) \approx x_1 \phi_1(t) + \cdots + x_d \phi_d(t)$$

A is an n-by-d "design matrix" (n >> d):

$$A_{ij} = \phi_j(t_i)$$

In matrix-vector notation,

$$y \approx Ax$$

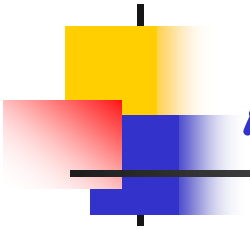
Least-norm approximation problems

The linear measurement model:

$$y = Ax + \varepsilon \quad \begin{cases} y \text{ are the measurements} \\ x \text{ is the unknown} \\ \varepsilon \text{ is an error process} \end{cases}$$

In order to estimate x, solve:

$$\widehat{x} = \arg\min \|y - Ax\|$$



Application: data analysis in science

- First application: Astronomy

  Predicting the orbit of the asteroid Ceres (in 1801!).

  Gauss (1809) -- see also Legendre (1805) and Adrain (1808).

  First application of "least squares optimization" and runs in $O(nd^2)$ time!

- Data analysis: Fit parameters of a biological, chemical, economical, physical, astronomical, social, internet, etc. model to experimental data.

Norms of common interest

Let y = b and define the residual: $r = Ax - b \in R^n$

Least-squares approximation:

$$\text{minimize: } ||Ax - b||_2^2 = r_1^2 + r_2^2 + \cdots + r_n^2$$

Chebyshev or mini-max approximation:

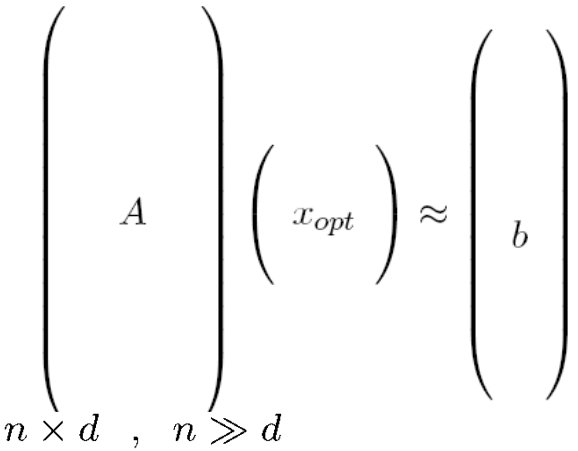
$$\text{minimize: } ||Ax - b||_\infty = \max\{|r_1|, \ldots, |r_n|\}$$

Sum of absolute residuals approximation:

$$\text{minimize: } ||Ax - b||_1 = |r_1| + |r_2| + \cdots + |r_n|$$

Least-squares problems

$$Z_2^2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2^2 = \|b - Ax_{opt}\|_2^2$$



We are interested in over-constrained least-squares problems, $n \gg d$.

We will briefly discuss under-constrained ($n \ll d$) and square ($n \approx d$) problems later.

Typically, there is no x_{opt} such that $Ax_{opt} = b$.

Want to find the "best" x_{opt} such that $Ax_{opt} \approx b$.

Exact solution to L_2 regression

Cholesky Decomposition:

If A is full rank and well-conditioned,

decompose $A^TA = R^TR$, where R is upper triangular, and

solve the normal equations: $R^TRx = A^Tb$.

QR Decomposition:

Slower but numerically stable, esp. if A is rank-deficient.

Write $A = QR$, and solve $Rx = Q^Tb$.

Singular Value Decomposition:

Most expensive, but best if A is very ill-conditioned.

Write $A = U\Sigma V^T$, in which case: $\mathbf{x_{opt} = A^+b = V\Sigma^{-1}U^Tb}$.

Complexity is $O(nd^2)$, but constant factors differ.

Projection of b on the subspace spanned by the columns of A

$$\mathcal{Z}_2^2 = \|b\|_2^2 - \left\|AA^+b\right\|_2^2$$

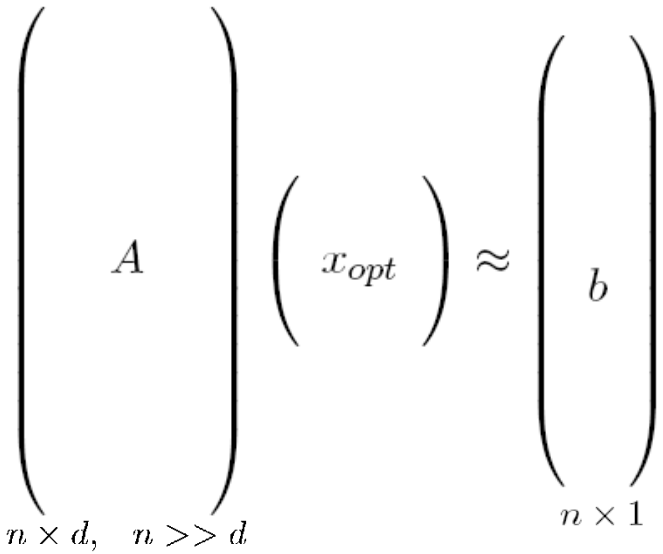
$$x_{opt} = A^+b$$

Pseudoinverse of A

Algorithm: Sampling for L_2 regression

(Drineas, Mahoney, Muthukrishnan SODA 2006, Sarlos FOCS 2007, Drineas, Mahoney, Muthukrishnan, & Sarlos NumMath2011)

$$\mathcal{Z}_2^2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2^2 = \|b - Ax_{opt}\|_2^2$$



Algorithm

1. Compute a probability distribution over the rows of A (p_i, i=1…n, summing up to one).
2. In r i.i.d. trials pick r rows of A and the corresponding elements of b with respect to the p_i.

   (Rescale sampled rows of A and sampled elements of b by $(1/(rp_i)^{1/2}$.)
3. Solve the induced problem.

Algorithm: Sampling for L_2 regression

(Drineas, Mahoney, Muthukrishnan SODA 2006, Sarlos FOCS 2007,
Drineas, Mahoney, Muthukrishnan, & Sarlos NumMath2011)

$$\tilde{Z}_2^2 = \min_{x \in \mathbb{R}^d} \left\| \tilde{b} - \tilde{A}x \right\|_2^2 = \left\| \tilde{b} - \tilde{A}\tilde{x}_{opt} \right\|_2^2$$

$$\underset{r \times d}{\begin{pmatrix} \\ \tilde{A} \\ \\ \end{pmatrix}} \begin{pmatrix} \\ \tilde{x}_{opt} \\ \\ \end{pmatrix} \approx \underset{r \times 1}{\begin{pmatrix} \\ \tilde{b} \\ \\ \end{pmatrix}}$$

Algorithm

1. Compute a probability distribution over the rows of A (p_i, i=1…n, summing up to one).
2. In r i.i.d. trials pick r rows of A and the corresponding elements of b with respect to the p_i.

   (Rescale sampled rows of A and sampled elements of b by $(1/(rp_i)^{1/2}$.)
3. Solve the induced problem.

Algorithm: Sampling for L_2 regression

(Drineas, Mahoney, Muthukrishnan SODA 2006, Sarlos FOCS 2007, Drineas, Mahoney, Muthukrishnan, & Sarlos NumMath2011)

$$\tilde{Z}_2^2 = \min_{x \in \mathbb{R}^d} \left\| \tilde{b} - \tilde{A}x \right\|_2^2 = \left\| \tilde{b} - \tilde{A}\tilde{x}_{opt} \right\|_2^2$$

$$\underset{r \times d}{\begin{pmatrix} \\ \tilde{A} \\ \\ \end{pmatrix}} \begin{pmatrix} \\ \tilde{x}_{opt} \\ \\ \end{pmatrix} \approx \underset{r \times 1}{\begin{pmatrix} \\ \tilde{b} \\ \\ \end{pmatrix}}$$

Algorithm

1. Compute a probability distribution over the rows of A (p_i, i=1…n, summing up to one).
2. In r i.i.d. trials pick r rows of A and the corresponding elements of b with respect to the p_i.

   (Rescale sampled rows of A and sampled elements of b by $(1/(rp_i)^{1/2}$.)
3. Solve the induced problem.

We will now discuss the p_i's: our work introduced the notion of the leverage scores.

Leverage scores: tall & thin matrices

Let A be a (full rank) n-by-d matrix with n>>d whose SVD is:

$$\underset{n \times d \ , \ n \gg d}{\begin{pmatrix} \\ A \\ \\ \end{pmatrix}} = \underset{n \times d}{\begin{pmatrix} \\ U \\ \\ \end{pmatrix}} \underset{d \times d}{\begin{pmatrix} \\ \Sigma \\ \\ \end{pmatrix}} \underset{d \times d}{\begin{pmatrix} \\ V^T \\ \\ \end{pmatrix}}$$

- The matrix U contains the left singular vectors of A.
- The columns of U are pairwise orthogonal and normal.
- This is NOT the case for rows of U: all we know is that the Euclidean norms of its rows are between zero and one.

Leverage scores: tall & thin matrices

Let A be a (full rank) n-by-d matrix with n>>d whose SVD is:

$$\underset{n \times d \ , \ n \gg d}{A} = \underset{n \times d}{U} \underset{d \times d}{\Sigma} \underset{d \times d}{V^T}$$

i-th row of U

(Row) Leverage scores: $p_i = \frac{\|U_{i*}\|_2^2}{\|U\|_F^2} = \frac{\|U_{i*}\|_2^2}{d}$

The (row) leverage scores can now be used to sample rows from A to create a sketch.

Computing leverage scores

Drineas, Magdon-Ismail, Mahoney, and Woodruff ICML 2012, JMLR 2012

- **Trivial:** via the Singular Value Decomposition

  $O(nd^2)$ time for n-by-d matrices with n>d.

- **Non-trivial:** relative error $(1+\varepsilon)$ approximations for all leverage scores.

Tall & thin matrices:

$$\begin{pmatrix} \\ A \\ \\ \end{pmatrix}$$

$n \times d \quad , \quad n \gg d$

Running time: $O(nd\varepsilon^{-2}\ \mathrm{polylog}(n/\varepsilon))$.

Theorem

If the p_i are the row leverage scores of A, then, with probability at least 0.8,

$$\|b - Ax_{opt}\|_2 \leq \|b - A\tilde{x}_{opt}\|_2 \leq (1+\epsilon)\ \|b - Ax_{opt}\|_2$$

The sampling complexity (the value of r) is

$$r = O\left(\frac{d}{\epsilon} + d \ln d\right)$$

Proof: a structural result

Consider the over-constrained least-squares problem:

$$\mathcal{Z}_2^2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2^2 = \|b - Ax_{opt}\|_2^2$$

and the "sketched" (or "preconditioned") problem

$$\tilde{\mathcal{Z}}_2^2 = \min_{x \in \mathbb{R}^d} \|X(b - Ax)\|_2^2 = \|Xb - XA\tilde{x}_{opt}\|_2^2$$

Recall: A is n-by-d with $n \gg d$; X is r-by-n with $r \ll n$.

- Think of XA as a "sketch" of A.
- Our approach (using the leverage scores) focused on sketches of A that consist of (rescaled) rows of A.
- More general matrices X are possible and have been heavily studied.

Proof: a structural result

$$\mathcal{Z}_2^2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2^2 = \|b - Ax_{opt}\|_2^2 \qquad \tilde{\mathcal{Z}}_2^2 = \min_{x \in \mathbb{R}^d} \|X(b - Ax)\|_2^2 = \|Xb - XA\tilde{x}_{opt}\|_2^2$$

Let U_A be the n-by-d matrix of the left singular vectors of A.

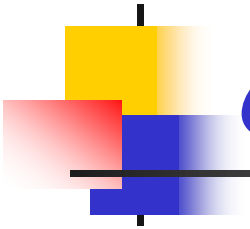
If X satisfies (constants are somewhat arbitrary):

$$b^{\perp} = b - U_A U_A^T b \qquad \sigma_{min}^2(XU_A) \geq 1/\sqrt{2}$$

$$\left\| U_A^T X^T X b^{\perp} \right\|_2^2 \leq \epsilon \mathcal{Z}_2^2/2,$$

then,

$$\begin{aligned} \|A\tilde{x}_{opt} - b\|_2 &\leq (1+\epsilon)\mathcal{Z}_2 \\ \|x_{opt} - \tilde{x}_{opt}\|_2 &\leq \frac{1}{\sigma_{min}(A)}\sqrt{\epsilon}\mathcal{Z}_2 \end{aligned}$$



Constructions for X

- If X is a sampling-and-rescaling matrix formed using the row leverage scores of the matrix A, then both conditions are satisfied.

  (I.e., an r-by-n matrix whose t-th row has a single non-zero entry indicating, and rescaling, the row of A that was sampled at the t-th trial.)

- Interestingly, many other matrices X satisfy both conditions: e.g., X can be a matrix whose entries are:
  - Random Gaussians (up to normalization).
  - Random signs (up to normalization).
  - The randomized Hadamard transform and its variants.
  - The input sparsity transform of Clarkson & Woodruff.

The "heart" of the proof

At the heart of all proofs in this line of research lies the following observation:

U_A is an orthogonal matrix: $U_A^T U_A = I_d$

$$\underbrace{\begin{pmatrix} \\ U_A \\ \\ \\ \end{pmatrix}}_{n \times d \quad , \quad n \gg d} \Rightarrow \underbrace{\begin{pmatrix} \\ XU_A \\ \\ \\ \end{pmatrix}}_{r \times d}$$

XU_A is a full-rank matrix!

$$r = O\left(\frac{d}{\epsilon^2} \ln\left(\frac{d}{\epsilon^2 \sqrt{\delta}}\right)\right)$$

Then, we can prove that with probability at least 1-δ:

$$\left\| U_A^T U_A - U_A^T X^T X U_A \right\|_2 = \left\| I - U_A^T X^T X U_A \right\|_2 \leq \varepsilon$$

It follows that, for all i: $\sqrt{1-\varepsilon} \leq \sigma_i \left(XU_A\right) \leq \sqrt{1+\varepsilon}$

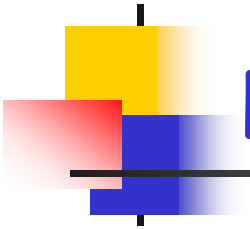
The "heart" of the proof (cont'd)

Recall: with probability at least 1-δ:

$$\left\|U_A^T U_A - U_A^T X^T X U_A\right\|_2 = \left\|I - U_A^T X^T X U_A\right\|_2 \leq \varepsilon$$

It follows that, for all i: $\sqrt{1-\varepsilon} \leq \sigma_i\left(XU_A\right) \leq \sqrt{1+\varepsilon}$

- The sampling complexity is r=O(d ln d).
- Proving the above inequality is (now) routinely done via matrix concentration inequalities (at least in most cases).
- Early proofs were very complicated and not user-friendly.



Follow-up

A lot of follow-up work, including:

- *Avron, Maymounkov, and Toledo SISC 2010:* Blendenpik, a solver that uses the "sketch" XA as a preconditioner, combined with an iterative least-squares solver. Beats LAPACK by a factor of four in essentially all over-constrained least-squares problems.
  - *Iyer, Avron, Kollias, Inechein, Carothers, and Drineas JCS 2016:* an evaluation of Blendenpik on terascale matrices in Rensselaer's BG/Q; again factor four-to-six speedups compared to Elemental's QR-based solver.
- *Drineas, Mahoney, Woodruff, and collaborators (SODA 2008, SIMAX 2009, SODA 2013, SIMAX 2016):* general p-norm regression, beyond Euclidean norm.
- *Clarkson and Woodruff STOC 2013:* relative error algorithms for over-constrained least-squares regression problems in input sparsity time using a novel construction for the sketching matrix X.

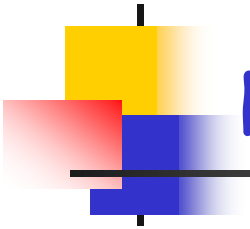
Follow-up (cont'd)

- *Pilanci and Wainwright IEEE TIF 2015, JMLR 2016, SIOPT 2017:* A novel iterative sketching-based method (Hessian sketch) to solve over-constrained least-squares regression problems over convex bodies.

- *Paul, Magdon-Ismail, and Drineas NIPS 2015, Derezinski and Warmuth NIPS 2017, AISTATS 2018, COLT 2018, JMLR 2018:* Adaptive and volume sampling approaches to construct the sketching matrix X.

- *Alaoui and Mahoney NIPS 2015, Cohen, Musco, Musco, and collaborators STOC 2015, SODA 2017, FOCS 2017:* ridge leverage scores, a smooth and regularized generalization of the leverage scores.

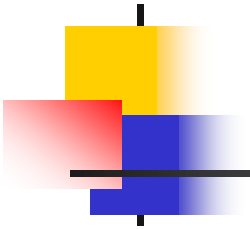
- *Chowdhuri, Yang, and Drineas ICML 2018:* structural conditions for under-constrained problems ($n \ll d$ case); a preconditioned Richardson-like solver for such problems; check our paper for a detailed discussion on prior work for such under-constrained problems.



Related work: the "square" case

The "square" case: solving systems of linear equations

- Almost optimal relative-error approximation algorithms for Laplacian and, more generally, Symmetric Diagonally Dominant (SDD) matrices
  - Pioneered by Spielman and Teng, major contributions later by Miller, Koutis, Peng, and many others.
  - Roughly speaking, the proposed methods are iterative preconditioned solvers where the preconditioner is a sparse version of the original graph.
  - This sparse graph is constructed by sampling edges of the original graph with probability proportional to their ***leverage scores***, which in the context of graphs are called ***effective resistances***.
- **Still open: progress beyond Laplacians.**
  - Results by Peng Zhang and Rasmus Kyng (FOCS 2017) indicate that such progress might be challenging.
- Check Koutis, Miler, and Peng CACM 2012 for a quick intro.



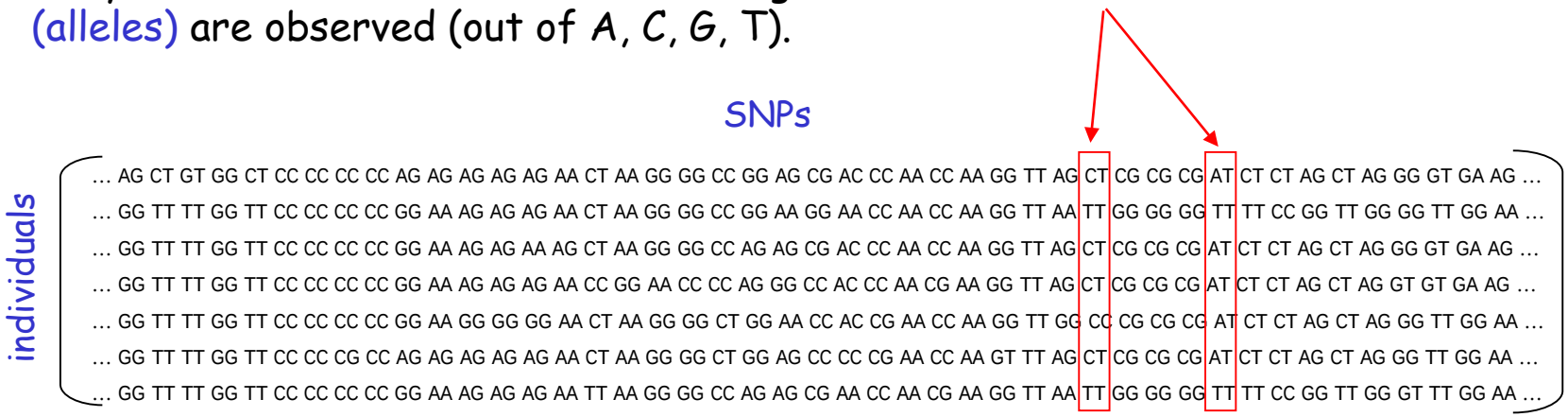
Roadmap

- RandNLA approaches for regression problems
- **RandNLA approaches for Principal Component Analysis (PCA)**

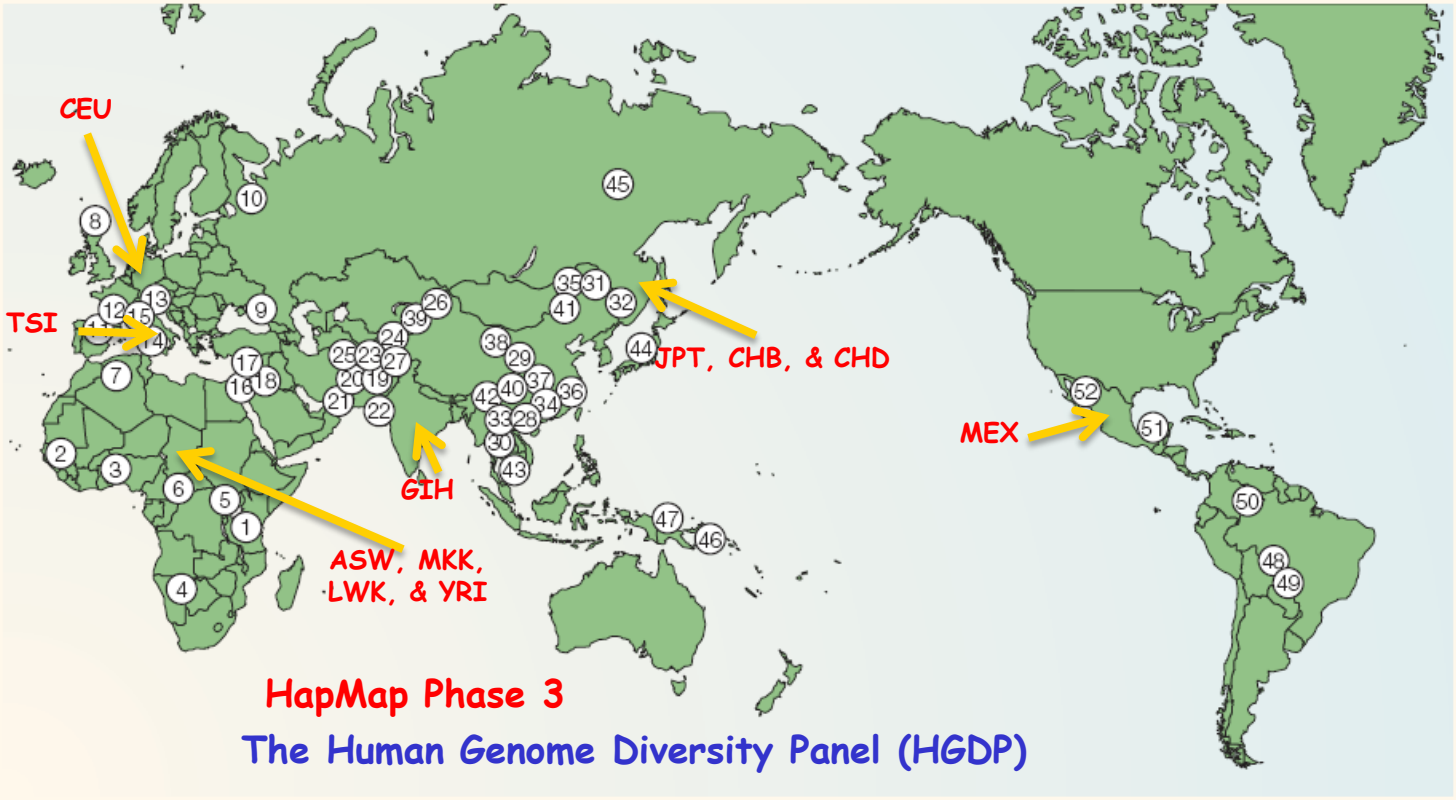
PCA: An example in human genetics

Single Nucleotide Polymorphisms: the most common type of genetic variation in the genome across different individuals.

They are known locations at the human genome where two alternate nucleotide bases (alleles) are observed (out of A, C, G, T).



Typical sizes: tens of thousands of individuals and hundreds of thousands of SNPs.



HGDP data

- 1,033 samples
- 7 geographic regions
- 52 populations

HapMap Phase 3 data

- 1,207 samples
- 11 populations

Matrix dimensions:

2,240 subjects (rows)

447,143 SNPs (columns)

Africans
1 Bantu
2 Mandenka
3 Yoruba
4 San
5 Mbuti pygmy
6 Biaka
7 Mozabite

Europeans
8 Orcadian
9 Adygei
10 Russian
11 Basque
12 French
13 North Italian
14 Sardinian
15 Tuscan

Western Asians
16 Bedouin
17 Druze
18 Palestinian

Central and Southern Asians
19 Balochi
20 Brahui
21 Makrani
22 Sindhi
23 Pathan
24 Burusho
25 Hazara
26 Uygur
27 Kalash

Eastern Asians
28 Han (S. China)
29 Han (N. China)
30 Dai
31 Daur
32 Hezhen
33 Lahu
34 Miao
35 Oroqen
36 She
37 Tujia
38 Tu
39 Xibo
40 Yi
41 Mongola
42 Naxi
43 Cambodian
44 Japanese
45 Yakut

Oceanians
46 Melanesian
47 Papuan

Native Americans
48 Karitiana
49 Surui
50 Colombian
51 Maya
52 Pima

Cavalli-Sforza (2005) *Nat Genet Rev*

Rosenberg et al. (2002) *Science*

Li et al. (2008) *Science*

The International HapMap Consortium (2003, 2005, 2007), *Nature*

We will apply PCA (i.e., SVD on a suitably rescaled covariance matrix) to visualize and/or analyze the data.

SVD: formal definition

$$\underset{m \times n}{\begin{pmatrix} & A & \end{pmatrix}} = \underset{m \times \rho}{\begin{pmatrix} & U & \end{pmatrix}} \cdot \underset{\rho \times \rho}{\begin{pmatrix} & \Sigma & \end{pmatrix}} \cdot \underset{\rho \times n}{\begin{pmatrix} & V & \end{pmatrix}^T}$$

ρ: rank of A

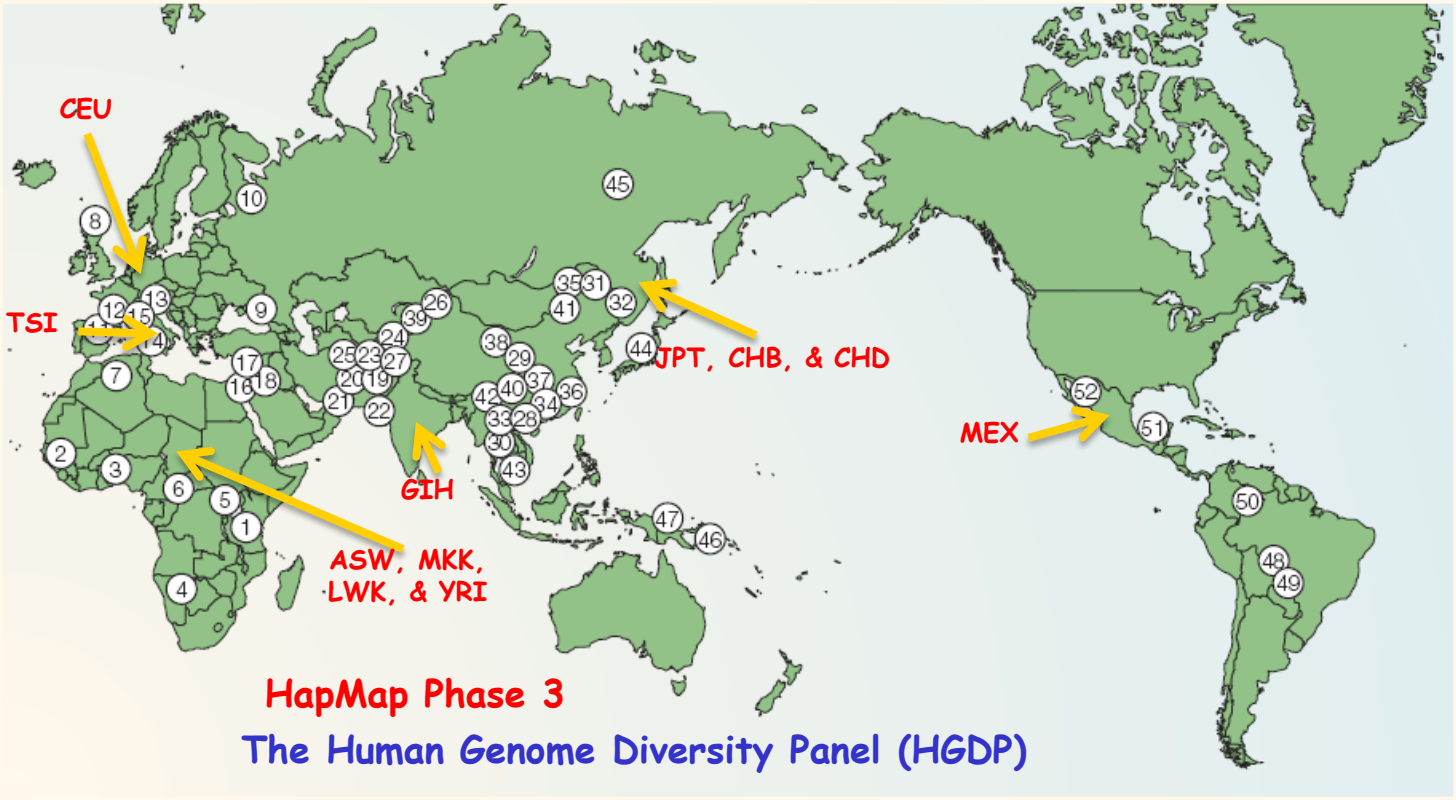
U (V): orthogonal matrix containing the left (right) singular vectors of A.

Σ: diagonal matrix containing the singular values of A.

Let σ_1 , σ_2 , … , σ_ρ be the entries of Σ.

Computing the SVD takes $O(\min\{mn^2 , m^2n\})$ time.

The top k left/right singular vectors/values can be computed faster using iterative methods.



HGDP data

- 1,033 samples
- 7 geographic regions
- 52 populations

HapMap Phase 3 data

- 1,207 samples
- 11 populations

Matrix dimensions:

2,240 subjects (rows)

447,143 SNPs (columns)

PCA on the above data returns:

Africans	Europeans	Western Asians	Eastern Asians	Oceanians
1 Bantu	8 Orcadian	16 Bedouin	28 Han (S. China)	46 Melanesian
2 Mandenka	9 Adygei	17 Druze	29 Han (N. China)	47 Papuan
3 Yoruba	10 Russian	18 Palestinian	30 Dai	
4 San	11 Basque		31 Daur	
5 Mbuti pygmy	12 French	Central and Southern Asians	32 Hezhen	Native Americans
6 Biaka	13 North Italian	19 Balochi	33 Lahu	48 Karitiana
7 Mozabite	14 Sardinian	20 Brahui	34 Miao	49 Surui
	15 Tuscan	21 Makrani	35 Oroqen	50 Colombian
		22 Sindhi	36 She	51 Maya
		23 Pathan	37 Tujia	52 Pima
		24 Burusho	38 Tu	
		25 Hazara	39 Xibo	
		26 Uygur	40 Yi	
		27 Kalash	41 Mongola	
			42 Naxi	
			43 Cambodian	
			44 Japanese	
			45 Yakut	

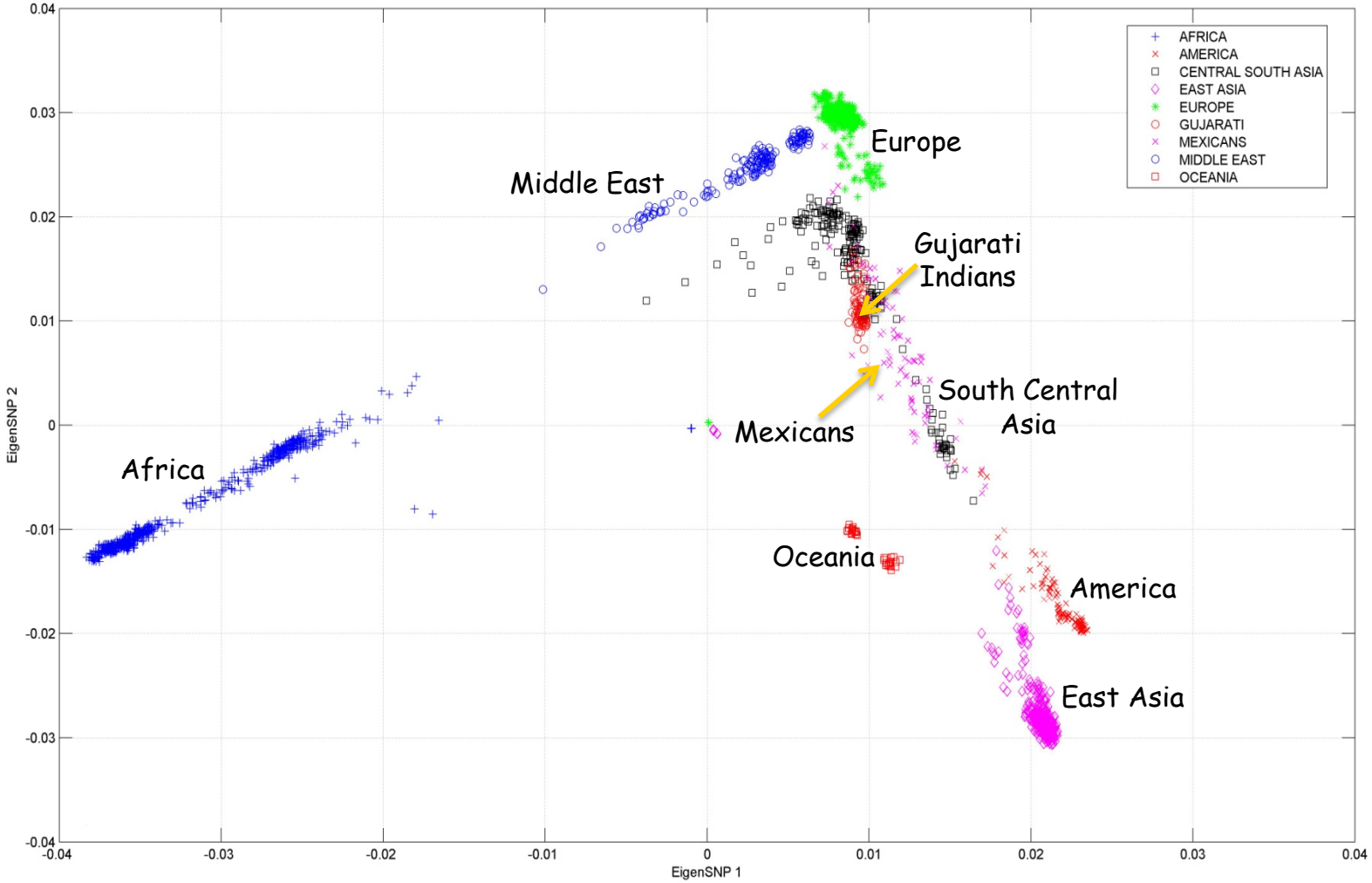
Cavalli-Sforza (2005) *Nat Genet Rev*

Rosenberg et al. (2002) *Science*

Li et al. (2008) *Science*

The International HapMap Consortium (2003, 2005, 2007), *Nature*

Paschou, Lewis, Javed, & Drineas (2010) J Med Genet

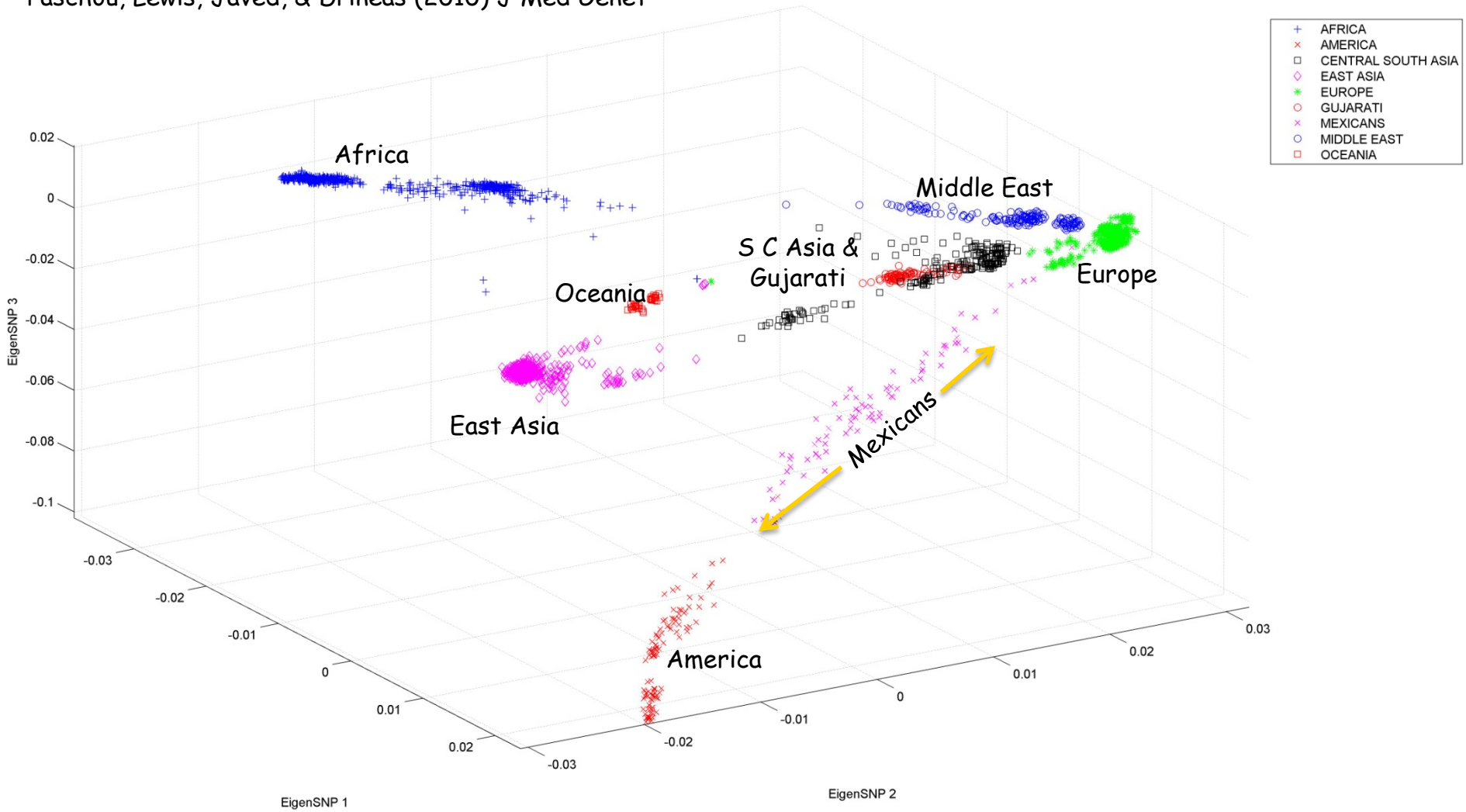


- Top two Principal Components (PCs or eigenSNPs)

(Lin and Altman (2005) *Am J Hum Genet*)

- Very good correlation between geography and the top two eigenSNPs.
- Mexican population seems out of place: we move to the top three PCs.

Paschou, Lewis, Javed, & Drineas (2010) J Med Genet

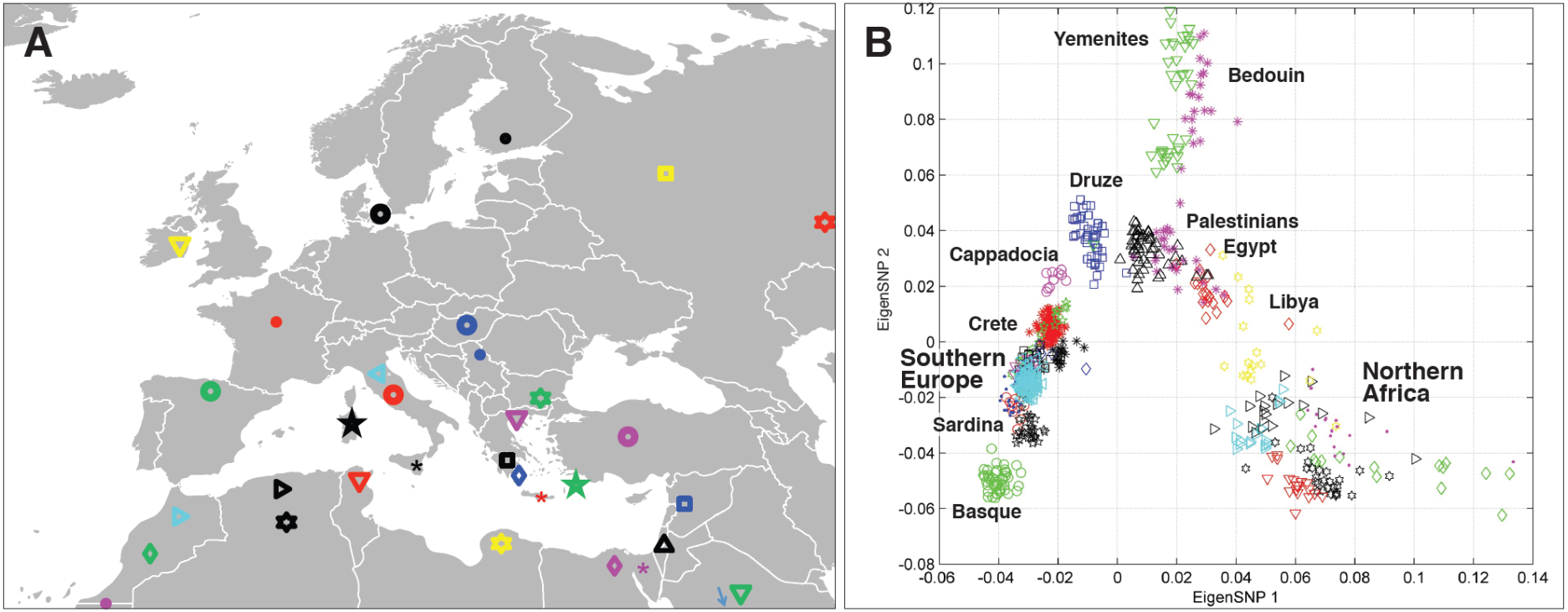


Not altogether satisfactory: the principal components are linear combinations of all SNPs, and – of course – can not be assayed!

Can we find **actual SNPs** that capture the information in the singular vectors?

Formally: spanning the same subspace.

Paschou, Drineas, et. al. (2014) PNAS



- PCA plots of genetic data from multiple populations around the Mediterranean Sea indicate that the Mediterranean acted as a **"barrier"** during the colonization of Europe from our species.
- Using PCA (and many other analyses) we proposed what is a called a **maritime route** for the colonization of Europe.
- Interpreting the singular vectors is, again, tricky; we identified SNPs (and genes) that capture the information in the singular vectors.

EJHG Open

European Journal of Human Genetics (2017), 1–9
Official journal of The European Society of Human Genetics
www.nature.com/ejhg

ARTICLE

Genetics of the peloponnesean populations and the theory of extinction of the medieval peloponnesean Greeks

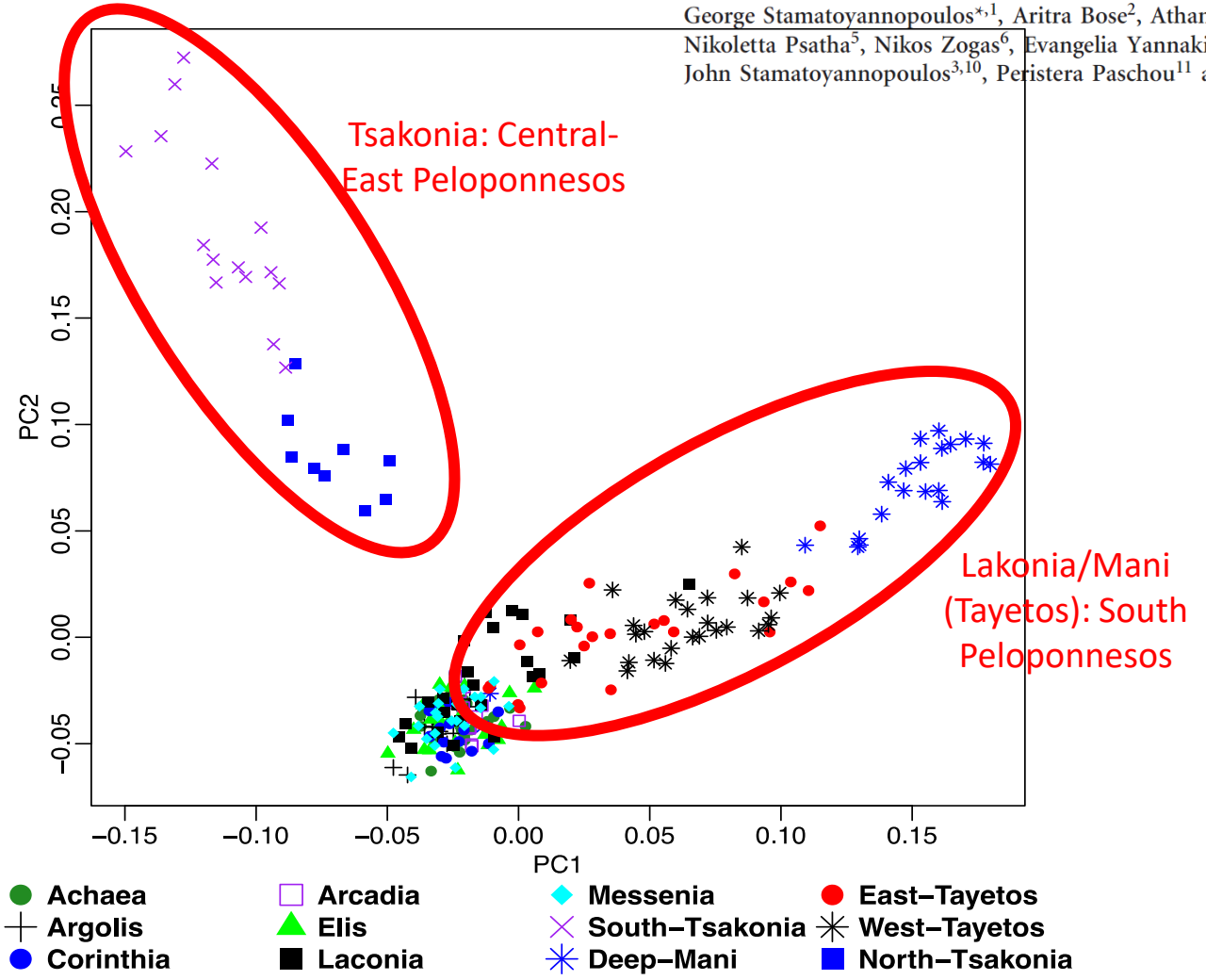
George Stamatoyannopoulos*,[1], Aritra Bose[2], Athanasios Teodosiadis[3], Fotis Tsetsos[2], Anna Plantinga[4], Nikoletta Psatha[5], Nikos Zogas[6], Evangelia Yannaki[6], Pierre Zalloua[7], Kenneth K Kidd[8], Brian L Browning[4,9], John Stamatoyannopoulos[3,10], Peristera Paschou[11] and Petros Drineas[2]

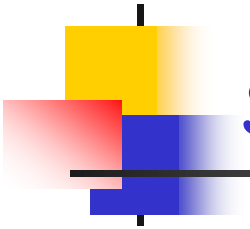
PCA identifies and extracts genetic micro-structure at very local levels and small geographical distances.

Consider, for example, Peloponnesos.

Genetics of the peloponnesean populations and the theory of extinction of the medieval peloponnesean Greeks

George Stamatoyannopoulos*[,1], Aritra Bose[2], Athanasios Teodosiadis[3], Fotis Tsetsos[2], Anna Plantinga[4], Nikoletta Psatha[5], Nikos Zogas[6], Evangelia Yannaki[6], Pierre Zalloua[7], Kenneth K Kidd[8], Brian L Browning[4,9], John Stamatoyannopoulos[3,10], Peristera Paschou[11] and Petros Drineas[2]





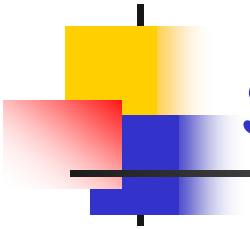
SVD: computational time

- **Computing large SVDs: computational time**
  - In commodity hardware (e.g., a 4GB RAM, dual-core laptop), using MatLab 7.0 (R14), the computation of the SVD of the dense 2,240-by-447,143 matrix A takes about 12 minutes.
  - Computing this SVD is not a one-liner, since we can not load the whole matrix in RAM (runs out-of-memory in MatLab); we compute the eigendecomposition of AA^T.
  - In 2010, we had to compute **1,200 SVDs** on matrices of dimensions (approx.) 1,200-by-450,000 for a full leave-one-out cross-validation experiment.

    (Drineas, Lewis, & Paschou (2010) PLoS ONE)
  - To compare mtDNA derived from 37 ancient Minoan bones to 120 extant and ancient populations we ran (multiple) SVDs on (approx.) 14,000-by-14,000 matrices.

    (Hughey, Paschou, Drineas, et al. (2013) Nat Comm)
  - Current population genetics datasets generate 1,000,000-by-1,000,000 matrices

    (Bose et al. (2018) TeraPCA package.)



SVD: computational time

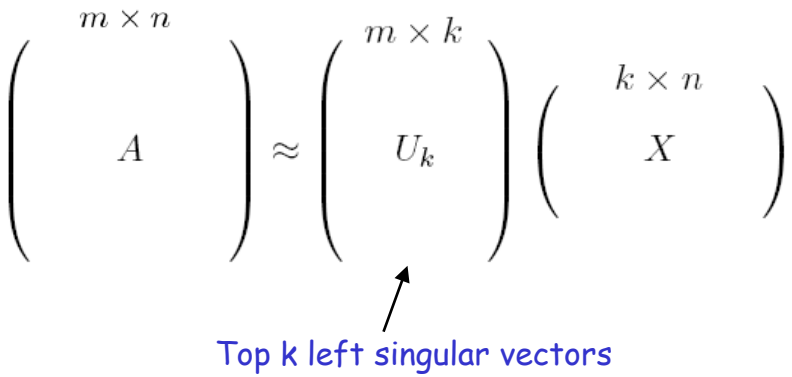
- **Computing large SVDs: computational time**
  - In commodity hardware (e.g., a 4GB RAM, dual-core laptop), using MatLab 7.0 (R14), the computation of the SVD of the dense 2,240-by-447,143 matrix A takes about 12 minutes.
  - Computing this SVD is not a one-liner, since we can not load the whole matrix in RAM (runs out-of-memory in MatLab); we compute the eigendecomposition of AA^T.
  - In 2010, we had to compute **1,200 SVDs** on matrices of dimensions (approx.) 1,200-by-450,000 for a full leave-one-out cross-validation experiment.

    (Drineas, Lewis, & Paschou (2010) PLoS ONE)
  - To compare mtDNA derived from 37 ancient Minoan bones to 120 extant and ancient populations we ran (multiple) SVDs on (approx.) 14,000-by-14,000 matrices.

    (Hughey, Paschou, Drineas, et al. (2013) Nat Comm)
  - Current population genetics datasets generate 1,000,000-by-1,000,000 matrices.

    (Bose et al. (2018) TeraPCA package.)
- **Running time is always a concern, but machine-precision is not necessary!**
  - Data are noisy and approximate singular vectors work well in many settings.

SVD decomposes a matrix as…

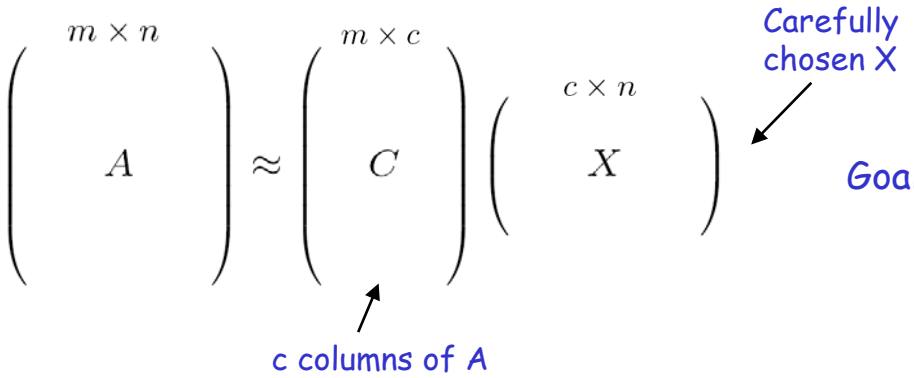


The SVD has strong optimality properties.

- It is easy to see that $X = \Sigma_k V_k^T = U_k^T A$.
- SVD has strong optimality properties.
- The columns of U_k are linear combinations of up to all columns of A.

The CX decomposition

Drineas, Mahoney, & Muthukrishnan (2008) SIAM J Mat Anal Appl
Mahoney & Drineas (2009) PNAS



Goal: make (some norm) of A-CX small.

Why?

If A is an subject-SNP matrix, then selecting representative columns is equivalent to selecting representative SNPs to capture the same structure as the top eigenSNPs.

We want c as small as possible!

CX decomposition

$$\underset{m \times n}{\begin{pmatrix} \\ A \\ \\ \end{pmatrix}} \approx \underset{m \times c}{\begin{pmatrix} \\ C \\ \\ \end{pmatrix}} \underset{c \times n}{\begin{pmatrix} \\ X \\ \\ \end{pmatrix}}$$

c columns of *A*

Easy to prove that optimal X = $C^{+}A$. (C^{+} is the Moore-Penrose pseudoinverse of *C*.)

Thus, the challenging part is to find good columns (SNPs) of *A* to include in *C*.

From a mathematical perspective, this is a hard combinatorial problem, closely related to the so-called Column Subset Selection Problem (CSSP).

The CSSP has been heavily studied in Numerical Linear Algebra.

Relative-error Frobenius norm bounds

Drineas, Mahoney, & Muthukrishnan (2008) SIAM J Mat Anal Appl

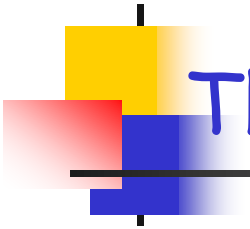
Given an m-by-n matrix A, there exists an $O(mn^2)$ algorithm that picks

at most $O((k/\varepsilon^2) \log (k/\varepsilon))$ columns of A

such that with probability at least .9

$$\|A - CX\|_F = \left\|A - CC^{\dagger}A\right\|_F \leq (1+\varepsilon)\ \|A - A_k\|_F$$

Notation: $\|X\|_F^2 = \sum_{i,j} X_{ij}^2$



The algorithm

Input: m-by-n matrix A,

$0 < \varepsilon < .5$, the desired accuracy

Output: C, the matrix consisting of the selected columns

Sampling algorithm

- Compute probabilities p_j summing to 1.
- Let $c = O((k/\varepsilon^2) \log (k/\varepsilon))$.
- In c i.i.d. trials pick columns of A, where in each trial the j-th column of A is picked with probability p_j.
- Let C be the matrix consisting of the chosen columns.

Subspace sampling (Frobenius norm)

$$\underset{m \times n}{\begin{pmatrix} \\ A_k \\ \\ \end{pmatrix}} = \underset{m \times k}{\begin{pmatrix} \\ U_k \\ \\ \end{pmatrix}} \cdot \underset{k \times k}{\begin{pmatrix} \\ \Sigma_k \\ \\ \end{pmatrix}} \cdot \underset{k \times n}{\begin{pmatrix} \\ V_k^T \\ \\ \end{pmatrix}}$$

V_k: orthogonal matrix containing the top k right singular vectors of A.

Σ_k: diagonal matrix containing the top k singular values of A.

Remark: The rows of V_k^T are orthonormal vectors, but its columns $(V_k^T)^{(i)}$ are not.

Leverage score sampling:

$$p_j = \frac{\left\| (V_k^T)^{(j)} \right\|_2^2}{k}$$

Normalization s.t. the p_j sum up to 1

Subspace sampling (Frobenius norm)

$$\underset{m \times n}{\left(A_k \right)} = \underset{m \times k}{\left(U_k \right)} \cdot \underset{k \times k}{\left(\Sigma_k \right)} \cdot \underset{k \times n}{\left(V_k^T \right)}$$

V_k: orthogonal matrix containing the top k right singular vectors of A.

Σ_k: diagonal matrix containing the top k singular values of A.

Remark: The rows of V_k^T are orthonormal vectors, but its columns $(V_k^T)^{(i)}$ are not.

Leverage score sampling:

Leverage scores

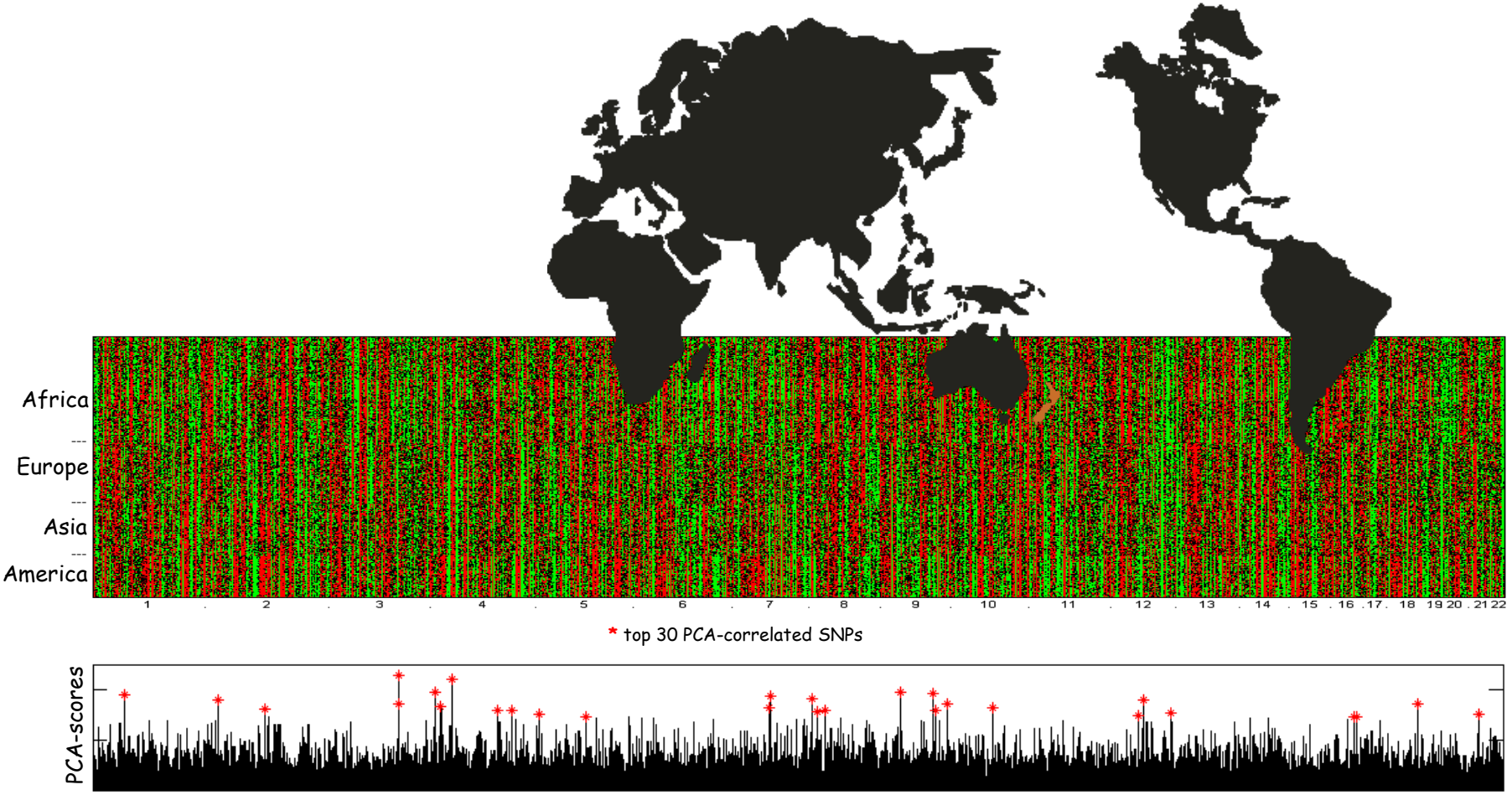
(useful in statistics for outlier detection)

$$p_j = \frac{\left\| \left(V_k^T \right)^{(j)} \right\|_2^2}{k}$$

Normalization s.t. the p_j sum up to 1

BACK TO POPULATION GENETICS DATA

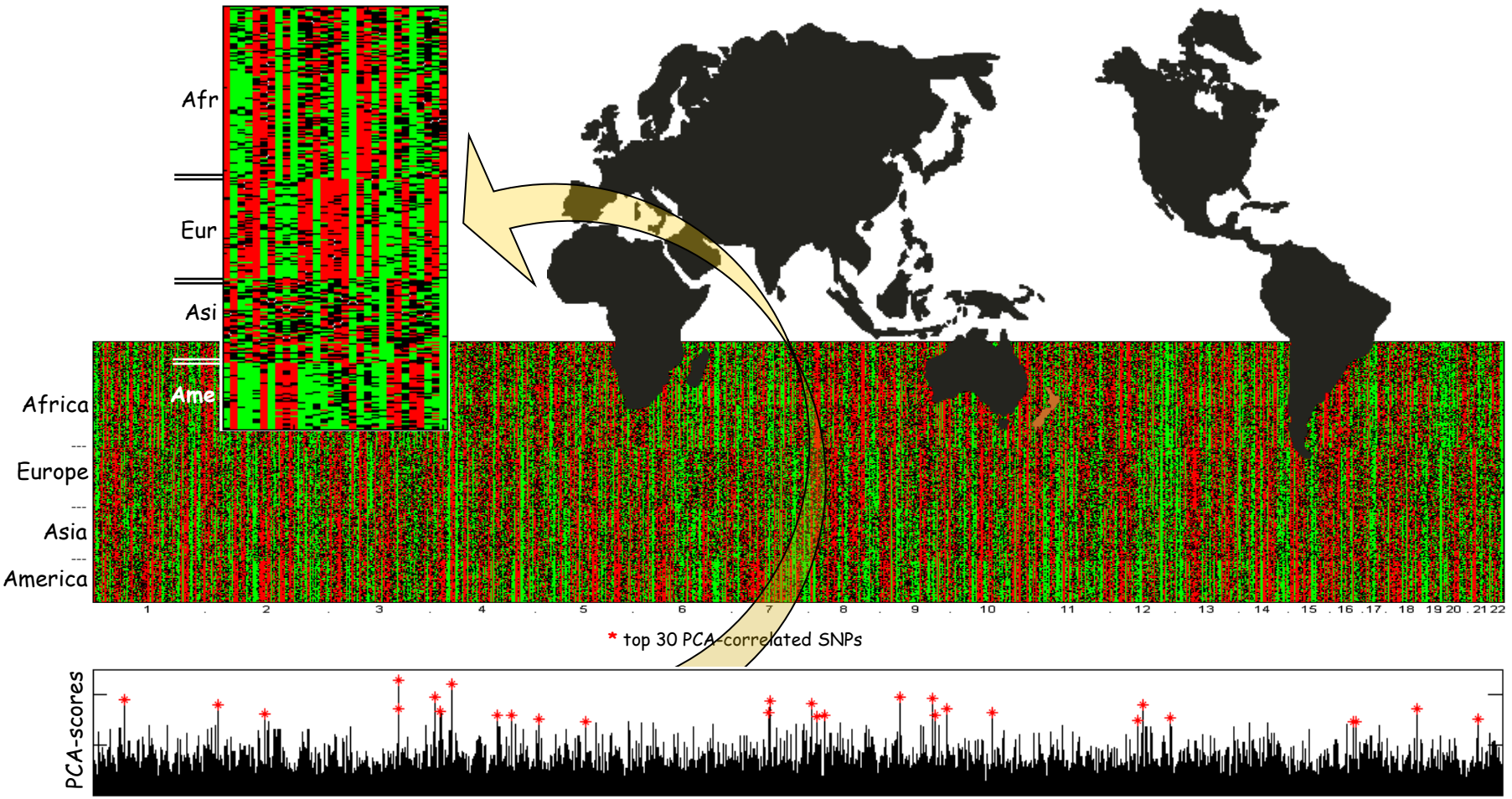
Selecting PCA SNPs for individual assignment to four continents (Africa, Europe, Asia, America)



Paschou et al (2007; 2008) PLoS Genetics; Paschou et al (2010) J Med Genet; Drineas et al (2010) PLoS One

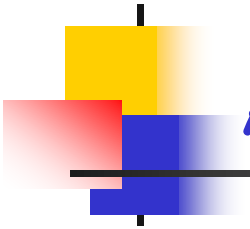
Hughey, Paschou, Drineas, et al. (2013) Nat Comm; Paschou, Drineas, et al. PNAS 2014;

Selecting PCA SNPs for individual assignment to four continents (Africa, Europe, Asia, America)



Paschou et al (2007; 2008) PLoS Genetics; Paschou et al (2010) J Med Genet; Drineas et al (2010) PLoS One

Hughey, Paschou, Drineas, et al. (2013) Nat Comm; Paschou, Drineas, et al. PNAS 2014;



Approximating leverage scores

Can we approximate the leverage scores fast?

Theorem: Given any m-by-n matrix A with $m > n$, we can approximate its leverage scores (where k is the target rank) with relative error accuracy in

$$O(mnk \log m) \text{ time},$$

as opposed to the – trivial – $O(mn^2)$ time.

(Drineas, Mahoney, Magdon-Ismail, & Woodruff ICML '12 JMLR '12)

Improvement: leverage scores can be computed in $O(nnz(A)\ k)$ time!

Clarkson and Woodruff (STOC '13): introduced a sparse random projection;

Mahoney and Meng (STOC '13): provided a better analysis for the above result;

Nelson and Huy (FOCS '13): provided the best known analysis for the above result;

Boutsidis and Woodruff (STOC '14): applications to many RandNLA problems.

Sobczyk and Gallopoulos '17: block iterative methods for fast estimation

Selecting fewer columns

Problem

How many columns do we need to include in the matrix C in order to get relative-error approximations ?

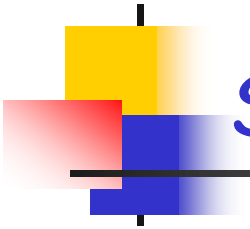
Recall: with $O((k/\varepsilon^2) \log (k/\varepsilon))$ columns, we get (subject to a failure probability)

$$\left\| A - CC^{\dagger}A \right\|_F \leq (1+\epsilon) \left\| A - A_k \right\|_F$$

Deshpande & Rademacher (FOCS '10): with exactly k columns, we get

$$\left\| A - CC^{\dagger}A \right\|_F \leq \sqrt{k} \left\| A - A_k \right\|_F$$

What about the range between k and O(k log(k))?



Selecting fewer columns (cont'd)

(Boutsidis, Drineas, & Magdon-Ismail, FOCS 2011 and SICOMP 2014)

Question:

What about the range between k and O(k log(k))?

Answer:

A relative-error bound is possible by selecting **$s=2k/\varepsilon$** columns!

Technical breakthrough;

A combination of sampling strategies with a novel approach on column selection, inspired by the work of Batson, Spielman, & Srivastava (STOC '09) on graph sparsifiers.

- **The running time is $O((mnk+nk^3)\varepsilon^{-1})$.**
- **Simplicity is gone…**

Lower bounds and alternative approaches

Deshpande & Vempala, RANDOM 2006

A relative-error approximation necessitates at **least k/ε columns.**

Guruswami & Sinop, SODA 2012

Alternative approaches, based on volume sampling, guarantee

$(r+1)/(r+1-k)$ relative error bounds.

This bound is asymptotically optimal (up to lower order terms).

The proposed deterministic algorithm runs in $O(rnm^3 \log m)$ time, while the randomized algorithm runs in $O(rnm^2)$ time and achieves the bound in expectation.

Guruswami & Sinop, FOCS 2011

Applications of column-based reconstruction in Quadratic Integer Programming.

Musco, Musco, Cohen, Woodruff, and collaborators

Multiple articles in STOC, FOCS, SODA, NIPS, ICML in 2016 and 2017 on ridge leverage scores and other approaches.

Iterative methods for PCA

(Drineas, Ipsen, Kontopoulou, and Magdon-Ismail SIMAX 2018
Drineas and Ipsen, under review SIMAX 2018)

To get highly accurate approximations to singular vectors, use iterative methods.

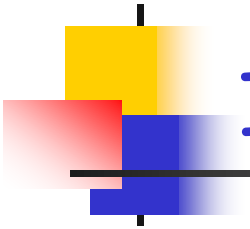
1. Block subspace iteration

Given an m-by-n matrix A and a positive integer q, compute

$$K = \left(AA^T\right)^q AX$$

where X is an n-by-p (with $p \approx k$) random matrix, e.g., a random Gaussian matrix.

Compute the best rank-k approximation to A within the subspace spanned by the columns of K (much easier to do than it sounds…): denote it by $\tilde{A}_k$.



Iterative methods for PCA (cont'd)

1. Block subspace iteration

Given an m-by-n matrix *A* and a positive integer q, compute

$$K = \left(AA^T\right)^q AX$$

where X is an n-by-p (with p ≈ k) random matrix, e.g., a random Gaussian matrix.

Compute the best rank-k approximation to *A* within the subspace spanned by the columns of K (much easier to do than it sounds…): denote it by $\tilde{A}_k$.

- Strong bounds can be proven for the Frobenius and spectral norms of the matrix $A - \tilde{A}_k$.
- We implemented block subspace iteration to approximate the top singular vectors of tera-scale matrices arising in population genetics in:

A. Bose, V. Kalantzis, E. Kontopoulou, M. Elkadi, P. Paschou, and P. Drineas, "*TeraPCA: a fast and scalable method to study genetic variation in tera-scale genotypes*", under review, Genome Biology, 2018.

Iterative methods for PCA

2. Block Krylov methods

Given an m-by-n matrix A (of rank ρ) and a positive integer q, compute

$$K = \left[AX,\ \left(AA^T\right) AX,\ \left(AA^T\right)^2 AX,\ \ldots\ , \left(AA^T\right)^q AX \right]$$

where X is an n-by-p (with $p \approx k$) random matrix, e.g., a random Gaussian matrix.

Compute the best rank-k approximation to A within the subspace spanned by the columns of K (much easier to do than it sounds): denote it by $\tilde{A}_k$.

- Assume a gap $g(>0)$ between the k and (k+1)-st singular values (can be relaxed):

$$\sigma_k \geq (1+g)\,\sigma_{k+1} > 0$$

Iterative methods for PCA

2. Block Krylov methods

Given an m-by-n matrix A (of rank ρ) and a positive integer q, compute

$$K = \left[AX,\ \left(AA^T\right)AX,\ \left(AA^T\right)^2 AX,\ \ldots\ ,\left(AA^T\right)^q AX\right]$$

where X is an n-by-p (with $p \approx k$) random matrix, e.g., a random Gaussian matrix.

Compute the best rank-k approximation to A within the subspace spanned by the columns of K (much easier to do than it sounds): denote it by $\tilde{A}_k$.

- Assume a gap $g(>0)$ between the k and $(k+1)$-st singular values (can be relaxed):

$$\sigma_k \geq (1+g)\,\sigma_{k+1} > 0$$

- Also assume (γ_1 and γ_2 are constants):

$$\sigma_{\min}^2\left(V_k^T X\right) \geq \gamma_1^2 \qquad \text{and} \qquad \left\|V_{k,\perp}^T X\right\|_F^2 \leq \gamma_2^2(\rho - k)$$

Bottom ρ-k singular vectors of A

Iterative methods for PCA

2. Block Krylov methods

Given an m-by-n matrix A (of rank ρ) and a positive integer q, compute

$$K = \left[AX,\ \left(AA^T\right)AX,\ \left(AA^T\right)^2 AX,\ \ldots\ ,\left(AA^T\right)^q AX \right]$$

$$q = O\left(\frac{\log(\rho/\epsilon)}{\sqrt{g}}\right)$$

where X is an n-by-p (with $p \approx k$) random matrix, e.g., a random Gaussian matrix.

Compute the best rank-k approximation to A within the subspace spanned by the columns of K (much easier to do than it sounds): denote it by $\tilde{A}_k$. Then,

$$\begin{aligned} \left\| A - \tilde{A}_k \right\|_F &\leq \|A - A_k\|_F + \epsilon\sigma_{k+1} \\ \left\| A - \tilde{A}_k \right\|_2 &\leq \|A - A_k\|_2 + \epsilon\sigma_{k+1} \end{aligned}$$

RandNLA and optimization

- Primal dual interior point methods necessitate solving least-squares problems (projecting the gradient on the null space of the constraint matrix in order to remain feasible).

  (Dating back to the mid/late 1980's and work by Karmarkar, Ye, Freund)

- Can we solve these least squares problems approximately using random sampling/random projections?
- **Modern approaches:** primal/dual interior point methods iterate along an approximation to the Newton direction and tolerate (mild) infeasibilities. A system of linear equations must be solved.

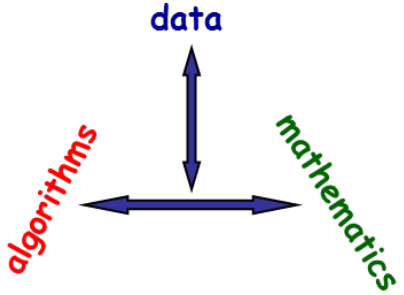
  (inexact interior point methods: work by Bellavia, Steihaug, etc.)

- **Well-known by practitioners:** the number of iterations in interior point methods is **not** the bottleneck, but the computational cost of solving a linear system is.
- **Goal:** Use sampling/random projection approaches to design efficient precoditioners to solve systems of linear equations that arise in primal-dual interior point methods faster.

Progress by Roosta & Mahoney (ArXiv 2016, 2017 multiple papers on subsampled second-order methods).

RandNLA events

"Randomization is arguably the most exciting and innovative idea to have hit linear algebra in a long time." (Avron et al. (2010) SISC)



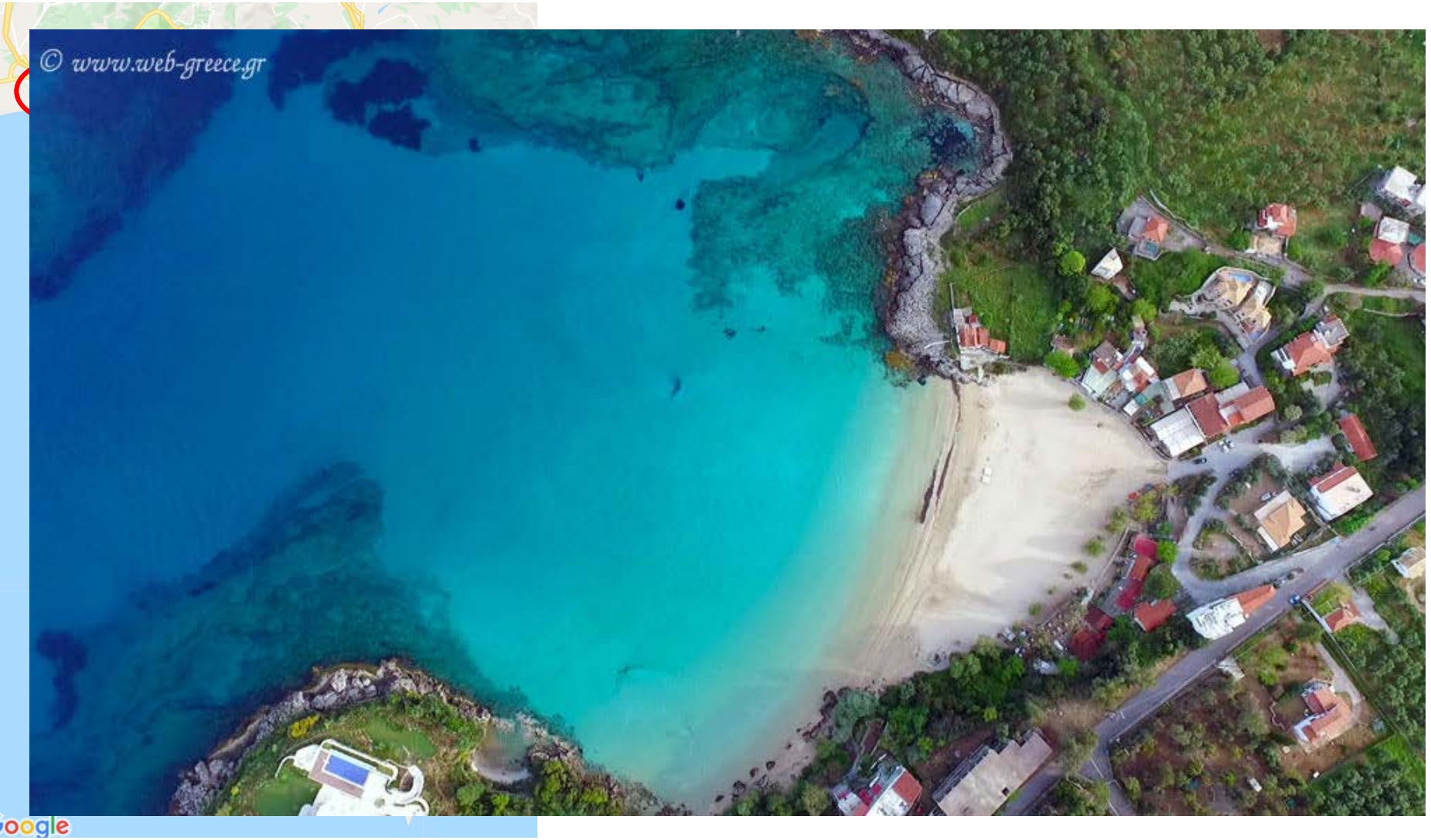
- RandNLA workshop, Simons Institute for the Theory of Computing, UC Berkeley, Foundations of Data Science, Sep 2018
  https://simons.berkeley.edu/data-science-2018-1
- RandNLA course, PCMI Summer School on Mathematics of Data, Jul 2016
  Drineas & Mahoney, Lectures on RandNLA, Vol. 25, Amer. Math. Soc., 2018
- Highlighted at the Workshops on Algorithms for Modern Massive Datasets (MMDS) 2006, 2008, 2010, 2012, 2014, and 2016.
  http://mmds-data.org/
- Gene Golub SIAM Summer School (G2S3), Δελφοί, Greece, June 2015
  http://scgroup19.ceid.upatras.gr/g2s32015/
- Invited tutorial at SIAM ALA 2015
- RandNLA workshop in FOCS 2012
  http://ieee-focs.org/focs2012/workshops/RandomNLA/

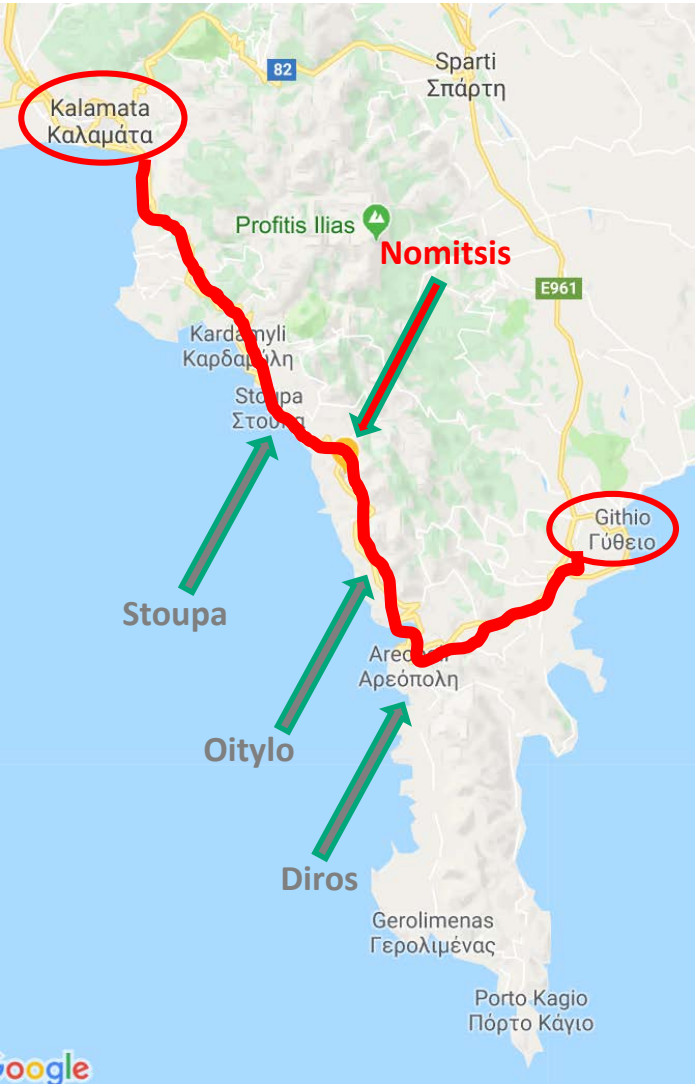
Very scenic drive full of the (old/restored/modern) tower-houses that Mani is famous for. Highlights:

Very scenic drive full of the (old/restored/modern) tower-houses that Mani is famous for. Highlights:

Stoupa

- Stoupa is where the (real) story of Zorba the Greek took place.


© www.web-greece.gr
oogle



Very scenic drive full of the (old/restored/modern) tower-houses that Mani is famous for. Highlights:

Stoupa

- Stoupa is where the (real) story of Zorba the Greek took place.

Nomitsis (this is where I come from…)

- Church of Agioi Anargyroi and Church of the Metamorphosis: a thousand years old.

n) tower-houses

ne Greek took place.

ne Metamorphosis: a


ΝΑΟΙ ΣΤΗ ΜΑΝΗ

Very scenic drive full of the (old/restored/modern) tower-houses that Mani is famous for. Highlights:

Stoupa

- Stoupa is where the (real) story of Zorba the Greek took place.

Nomitsis (this is where I come from…)

- Church of Agioi Anargyroi and Church of the Metamorphosis: a thousand years old.

Oitylo

- First mentioned in Homer's Iliad.

l/modern) tower-houses

f Zorba the Greek took place.

urch of the Metamorphosis: a

Very scenic drive full of the (old/restored/modern) tower-houses that Mani is famous for. Highlights:

Stoupa

- Stoupa is where the (real) story of Zorba the Greek took place.

Nomitsis (this is where I come from…)

- Church of Agioi Anargyroi and Church of the Metamorphosis: a thousand years old.

Oitylo

- First mentioned in Homer's Iliad.

Diros

- Diros caves: partially submerged an underground river.

ace.

is: a