SQUiD: Ultra-Secure Storage and Analysis of Genetic Data for the Advancement of Precision Medicine

Jacob Blindenbach[1,2,5], Jiayi Kang[3,5], Seungwan Hong[2,4,5], Caline Karam[2], Thomas Lehner[2], Gamze Gürsoy[4,2,1]

[1]Department of Computer Science, Columbia University; [2]New York Genome Center; [3]IMEC-COSIC, KU Leuven; [4]Department of Biomedical Informatics, Columbia University

[5]These authors contributed equally: Jacob Blindenbach, Jiayi Kang, and Seungwan Hong

COLUMBIA UNIVERSITY

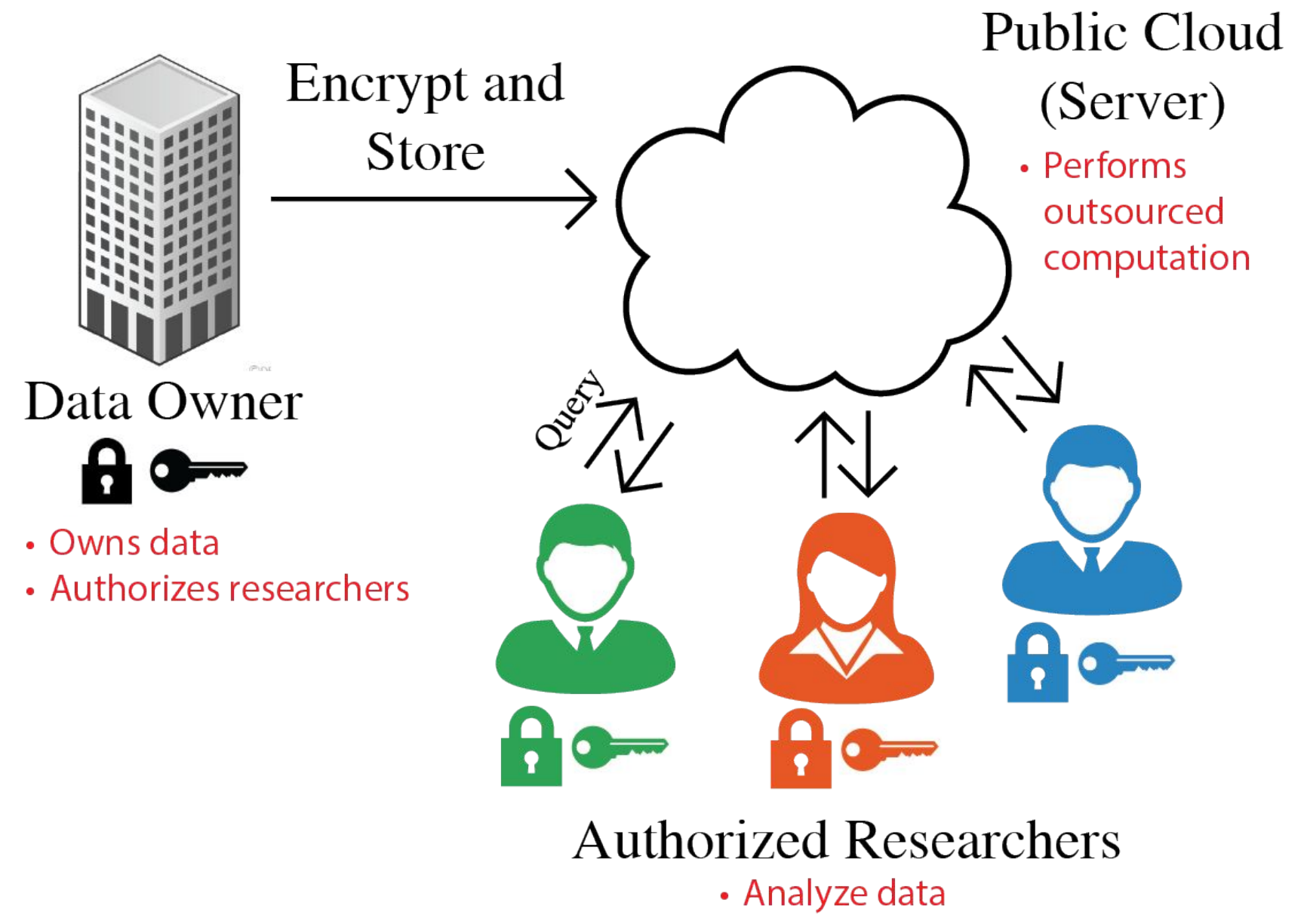
NEW YORK GENOME CENTER®

Problem Statement and Goal

- Queryable databases are needed for storing extensive, sensitive patient disease, and genetic information
- Large amount of data necessitates cloud storage, which necessitates strong security measures due to its sensitive nature

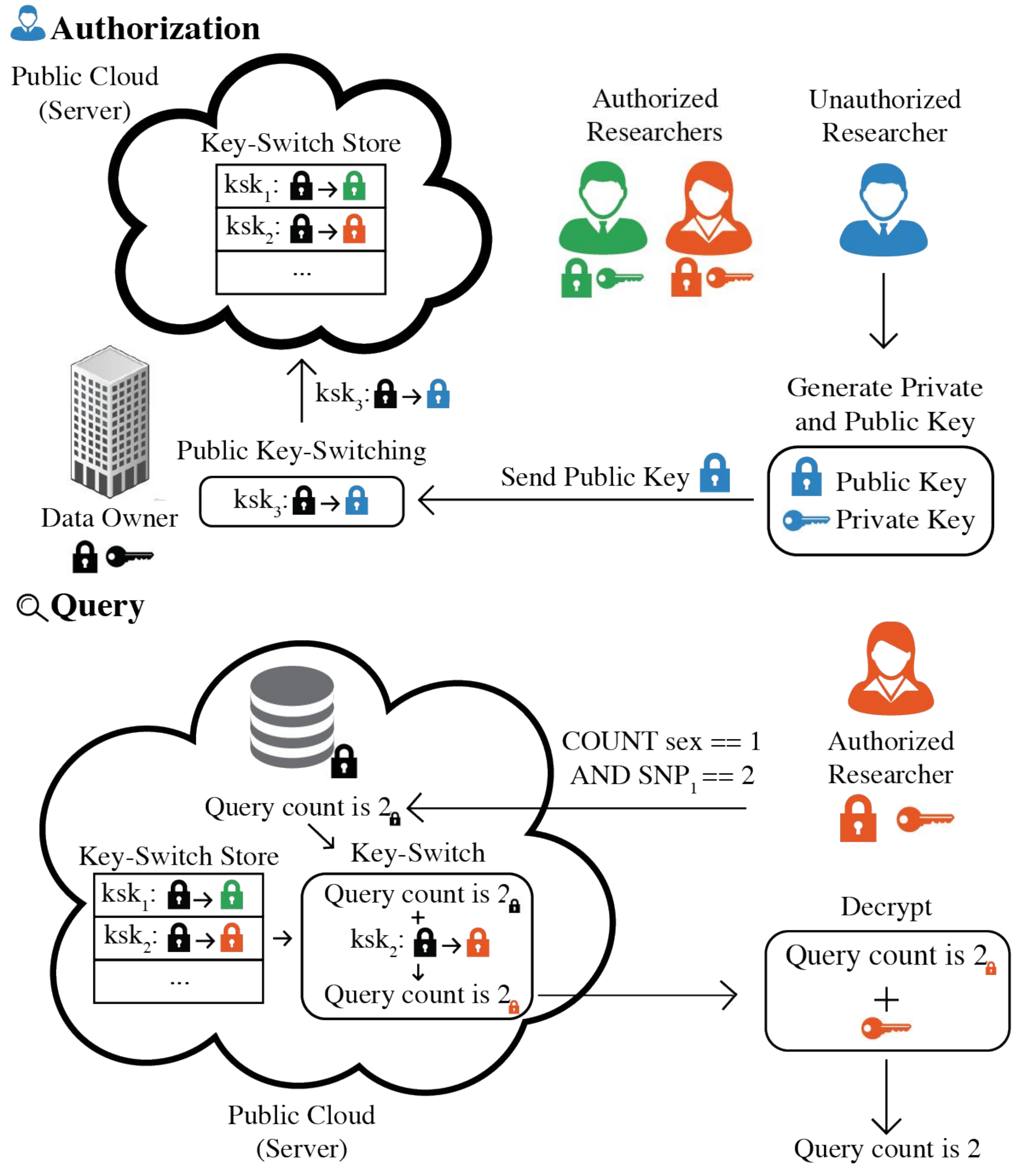
Goal: Ensure the data is secure from both cloud vulnerabilities and unauthorized users, yet accessible for authorized researchers to safely perform queries and analyses

Scenario



- **SQUiD** (**S**ecure **QU**eryable genotype-phenotype **D**atabases) is designed for a multiparty setting with a data owner, a public cloud, and multiple researchers
- SQUiD utilizes homomorphic encryption (HE) to securely compute on the data, which can be stored in the public cloud

Enabling multiparty queries with public key-switching



- We developed a new cryptographic primitive, "public key-switching", to eliminate the sharing and sending of private keys
- Each party generates their own set of keys:

  Public Key
  Private Key

- Only the public key is sent to the data owner to produce a key-switching key
- Key-switching re-encrypts a ciphertext, *ct* into a ciphertext, *ct'* such that the client can decrypt *ct'* with their private key
- Having a key-switching key inside the store, grants the researcher access to the data

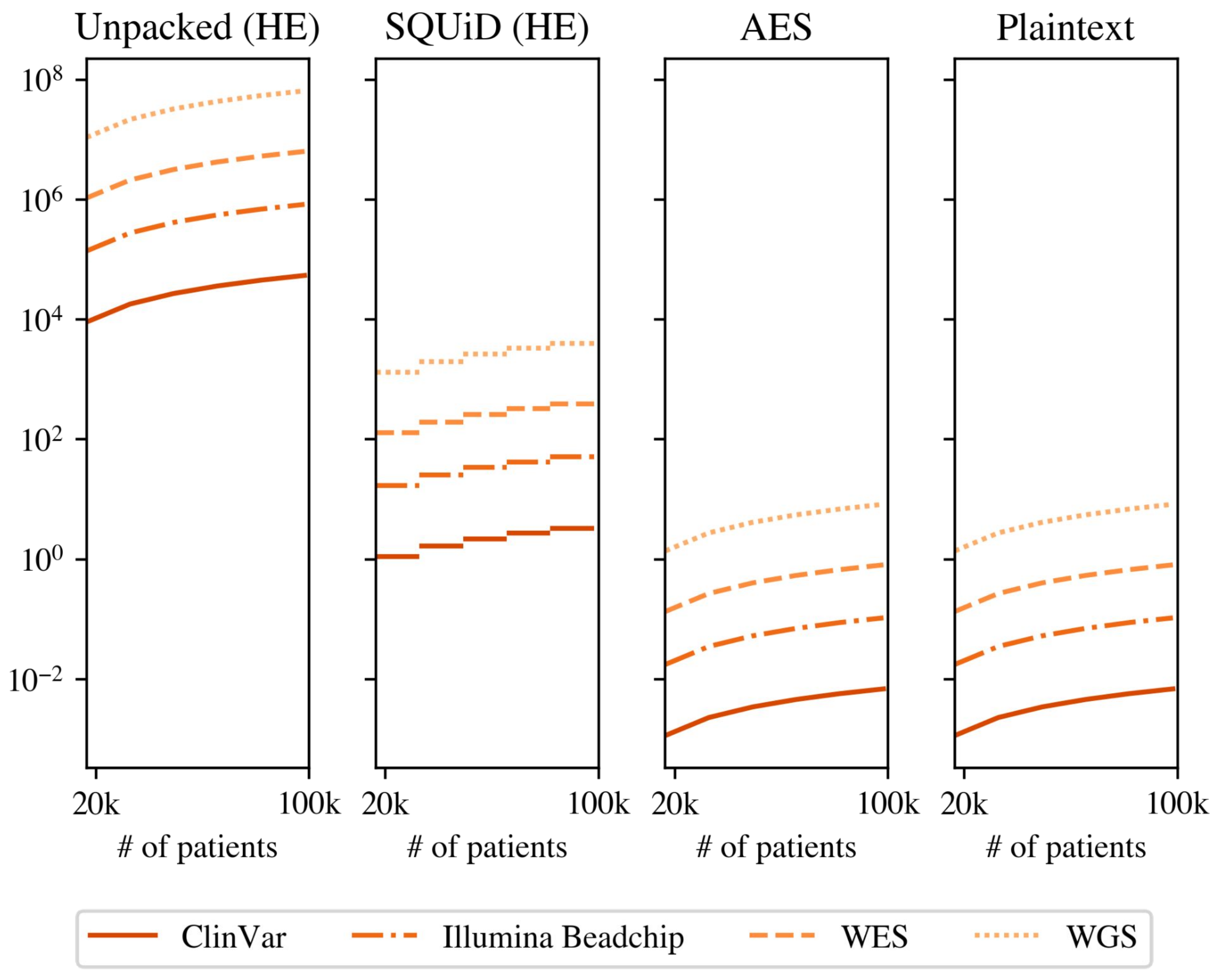
Overcoming high storage costs by utilizing vertical packing

- SQUiD vertically packs columns of genotypes and phenotypes from multiple individuals into single ciphertexts

ID	SNP$_1$	SNP$_2$	⋯	SNP$_K$	sex	smoker	disease
0	0	1	⋯	1	0	0	3
1	1	0	⋯	0	1	1	2
2	0	1	⋯	0	1	0	3
⋮	⋮	⋮		⋮	⋮	⋮	⋮
N	0	0	⋯	2	0	1	0

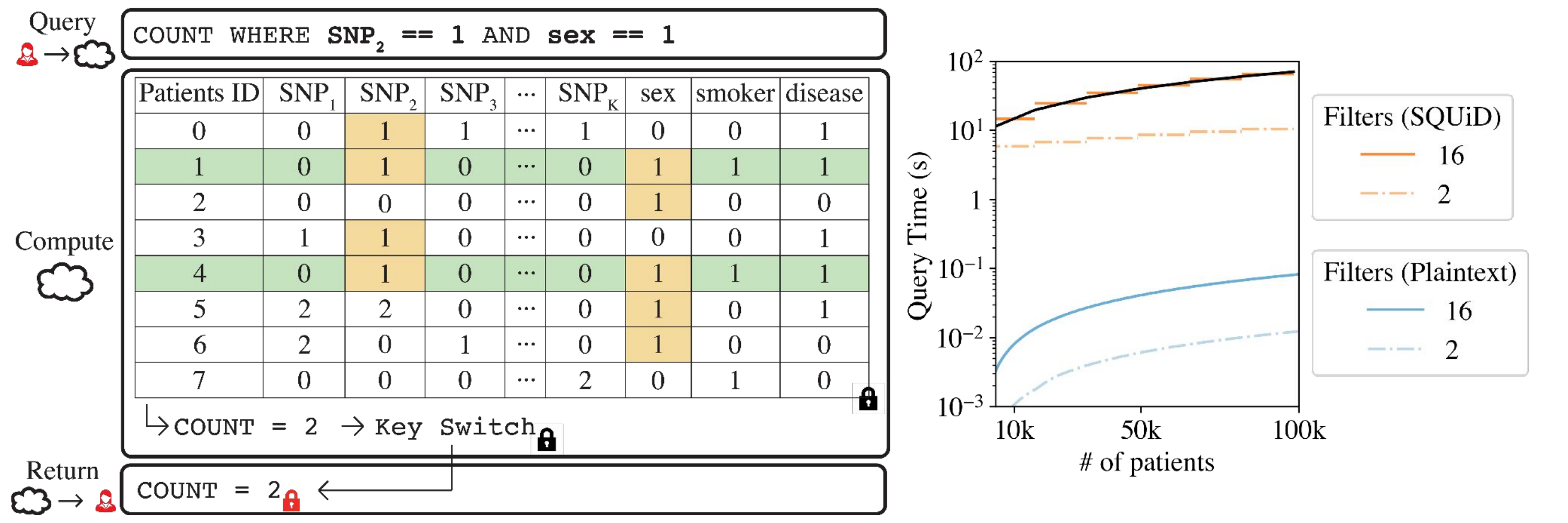
Columns are packed into single ciphertext

ID	SNP$_1$	SNP$_2$	⋯	SNP$_K$	sex	smoker	disease
0	c_0	c_1	⋯	c_{K-1}	c_K	c_{K+1}	c_{K+2}
1							
2							
⋮							
N							



Providing scalable query performance

A - Count Query



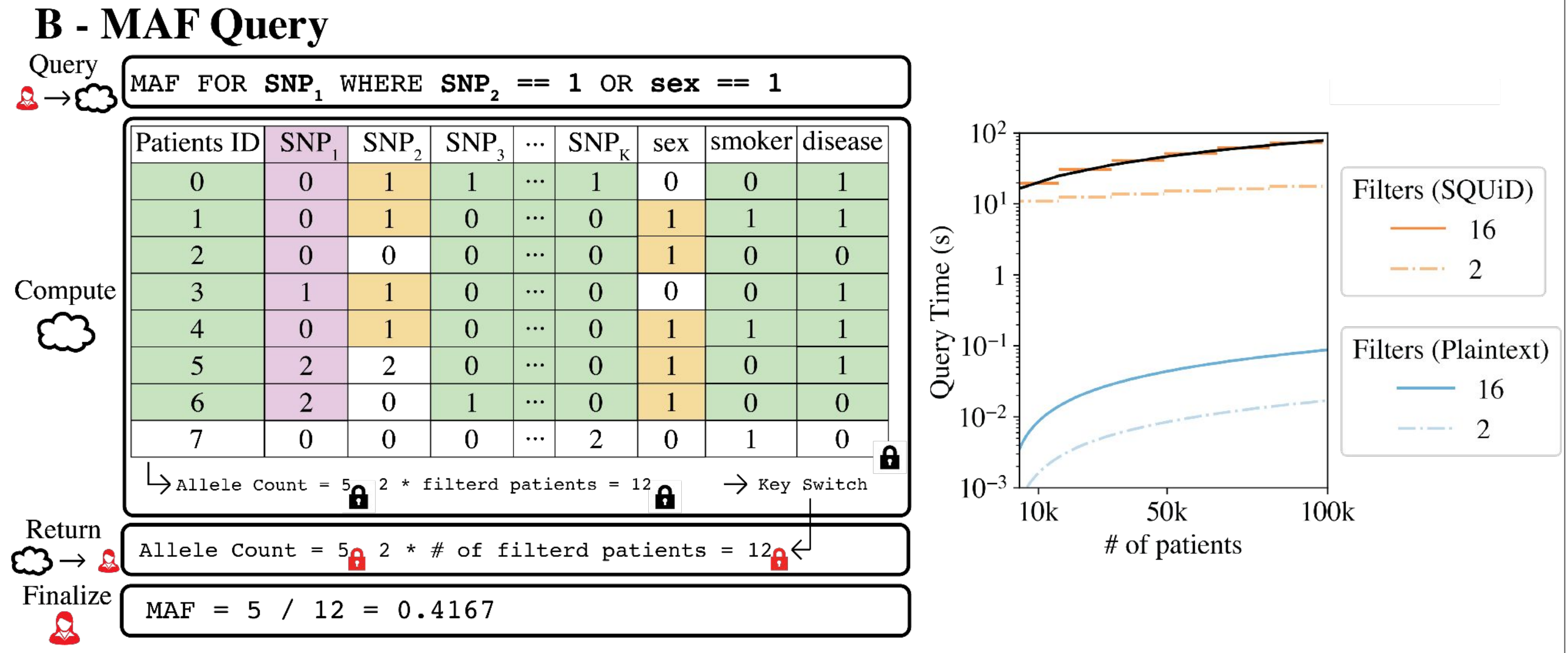
COUNT WHERE **SNP$_2$** == 1 AND **sex** == 1

Patients ID	SNP$_1$	SNP$_2$	SNP$_3$	⋯	SNP$_K$	sex	smoker	disease
0	0	1	1	⋯	1	0	0	1
1	0	1	0	⋯	0	1	1	1
2	0	0	0	⋯	0	1	0	0
3	1	1	0	⋯	0	0	0	1
4	0	1	0	⋯	0	1	1	1
5	2	2	0	⋯	0	1	0	1
6	2	0	1	⋯	0	1	0	0
7	0	0	0	⋯	2	0	1	0

↳COUNT = 2 → Key Switch

COUNT = 2 ←

B - MAF Query



MAF FOR **SNP$_1$** WHERE **SNP$_2$** == 1 OR **sex** == 1

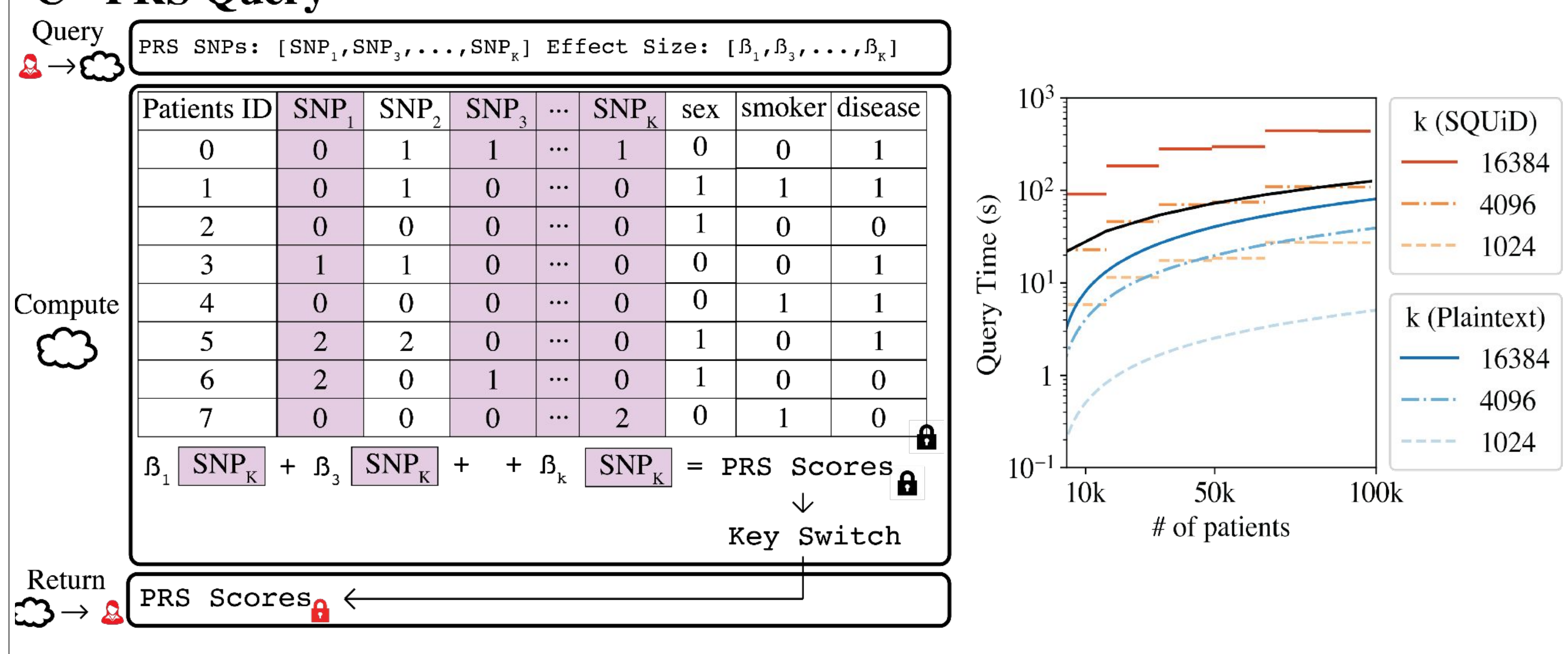
Patients ID	SNP$_1$	SNP$_2$	SNP$_3$	⋯	SNP$_K$	sex	smoker	disease
0	0	1	1	⋯	1	0	0	1
1	0	1	0	⋯	0	1	1	1
2	0	0	0	⋯	0	1	0	0
3	1	1	0	⋯	0	0	0	1
4	0	1	0	⋯	0	1	1	1
5	2	2	0	⋯	0	1	0	1
6	2	0	1	⋯	0	1	0	0
7	0	0	0	⋯	2	0	1	0

↳Allele Count = 5 2 * filterd patients = 12 → Key Switch

Allele Count = 5 2 * # of filterd patients = 12 ←

MAF = 5 / 12 = 0.4167

C - PRS Query



PRS SNPs: [SNP$_1$,SNP$_3$,...,SNP$_K$] Effect Size: [ß$_1$,ß$_3$,...,ß$_K$]

Patients ID	SNP$_1$	SNP$_2$	SNP$_3$	⋯	SNP$_K$	sex	smoker	disease
0	0	1	1	⋯	1	0	0	1
1	0	1	0	⋯	0	1	1	1
2	0	0	0	⋯	0	1	0	0
3	1	1	0	⋯	0	0	0	1
4	0	0	0	⋯	0	0	1	1
5	2	2	0	⋯	0	1	0	1
6	2	0	1	⋯	0	1	0	0
7	0	0	0	⋯	2	0	1	0

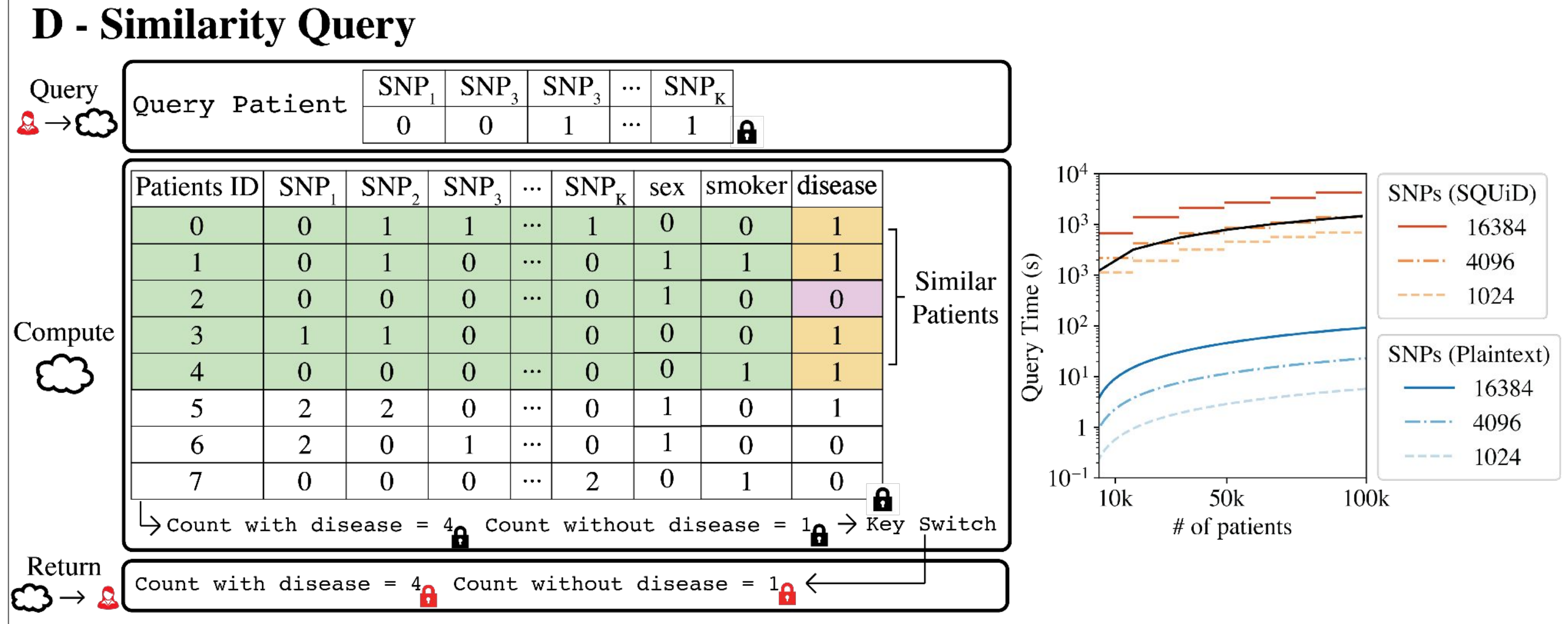
ß$_1$ SNP$_K$ + ß$_3$ SNP$_K$ + + ß$_k$ SNP$_K$ = PRS Scores

↓

Key Switch

PRS Scores ←

D - Similarity Query



Query Patient

SNP$_1$	SNP$_3$	SNP$_3$	⋯	SNP$_K$
0	0	1	⋯	1

Patients ID	SNP$_1$	SNP$_2$	SNP$_3$	⋯	SNP$_K$	sex	smoker	disease
0	0	1	1	⋯	1	0	0	1
1	0	1	0	⋯	0	1	1	1
2	0	0	0	⋯	0	1	0	0
3	1	1	0		0	0	0	1
4	0	0	0	⋯	0	0	1	1
5	2	2	0	⋯	0	1	0	1
6	2	0	1	⋯	0	1	0	0
7	0	0	0	⋯	2	0	1	0

↳Count with disease = 4 Count without disease = 1 → Key Switch

Count with disease = 4 Count without disease = 1 ←

Acknowledgements: This work has been funded by NIH grants R00HG010909 and R35GM147004 to GG and Warren Alpert Foundation and WoodNext Foundation awards to TL.

Github repository of SQUiD with an API for quick deployment: