PLOS | COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Convex Clustering: An Attractive Alternative to Hierarchical Clustering

Gary K. Chen[1]*, Eric C. Chi[2], John Michael O. Ranola[3], Kenneth Lange[4,5,6]

1 Department of Preventive Medicine, Biostatistics Division, University of Southern California, Los Angeles, California, United States of America, **2** Department of Electrical and Computer Engineering, Rice University, Houston, Texas, United States of America, **3** Department of Statistics, University of Washington, Seattle, Washington, United States of America, **4** Department of Biomathematics, University of California, Los Angeles, Los Angeles, California, United States of America, **5** Department of Human Genetics, University of California, Los Angeles, Los Angeles, California, United States of America, **6** Department of Statistics, University of California, Los Angeles, Los Angeles, California, United States of America

* gary.k.chen@usc.edu

Abstract

The primary goal in cluster analysis is to discover natural groupings of objects. The field of cluster analysis is crowded with diverse methods that make special assumptions about data and address different scientific aims. Despite its shortcomings in accuracy, hierarchical clustering is the dominant clustering method in bioinformatics. Biologists find the trees constructed by hierarchical clustering visually appealing and in tune with their evolutionary perspective. Hierarchical clustering operates on multiple scales simultaneously. This is essential, for instance, in transcriptome data, where one may be interested in making qualitative inferences about how lower-order relationships like gene modules lead to higher-order relationships like pathways or biological processes. The recently developed method of convex clustering preserves the visual appeal of hierarchical clustering while ameliorating its propensity to make false inferences in the presence of outliers and noise. The solution paths generated by convex clustering reveal relationships between clusters that are hidden by static methods such as k-means clustering. The current paper derives and tests a novel proximal distance algorithm for minimizing the objective function of convex clustering. The algorithm separates parameters, accommodates missing data, and supports prior information on relationships. Our program CONVEXCLUSTER incorporating the algorithm is implemented on ATI and nVidia graphics processing units (GPUs) for maximal speed. Several biological examples illustrate the strengths of convex clustering and the ability of the proximal distance algorithm to handle high-dimensional problems. CONVEXCLUSTER can be freely downloaded from the UCLA Human Genetics web site at http://www.genetics.ucla.edu/software/

Author Summary

Pattern discovery is one of the most important goals of data-driven research. In the biological sciences hierarchical clustering has achieved a position of pre-eminence due to its

OPEN ACCESS

Citation: Chen GK, Chi EC, Ranola JMO, Lange K (2015) Convex Clustering: An Attractive Alternative to Hierarchical Clustering. PLoS Comput Biol 11(5): e1004228. doi:10.1371/journal.pcbi.1004228

Editor: Sergei L. Kosakovsky Pond, University of California San Diego, UNITED STATES

Received: August 27, 2014

Accepted: March 6, 2015

Published: May 12, 2015

Data Availability Statement: Datasets in paper are held in public repositories 1) https://archive.ics.uci.edu/ml/datasets.html 2) http://web.stanford.edu/group/rosenberglab/diversity.html 3) http://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id = phs000145.v4.p2 4) http://genome-www.stanford.edu/sutech/

Funding: This work was funded in part by National Institutes of Health grants R01 MH100879, R01 ES019876, and R01 HG006465 supporting GKC, CIA Postdoctoral Fellowship 2012-12062800003 supporting ECC, NHGRI grant T32 HG00035 supporting JMOR, and R01 HG006139 and RO1 GM53275 supporting KL. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

ability to capture multiple levels of data granularity. Hierarchical clustering's visual displays of phylogenetic trees and gene-expression modules are indeed seductive. Despite its merits, hierarchical clustering is greedy by nature and often produces spurious clusters, particularly in the presence of substantial noise. This paper presents a relatively new alternative to hierarchical clustering known as convex clustering. Although convex clustering is more computationally demanding, it enjoys several advantages over hierarchical clustering and other traditional methods of clustering. Convex clustering delivers a uniquely defined clustering path that partially obviates the need for choosing an optimal number of clusters. Along the path small clusters gradually coalesce to form larger clusters. Clustering can be guided by external information through appropriately defined similarity weights. Comparisons to hierarchical clustering demonstrate the superior robustness of convex clustering to noise. Our genetics examples include inference of the demographic history of 52 populations across the world, a more detailed analysis of European demography, and a re-analysis of a well-known breast cancer expression dataset. We also introduce a new algorithm for solving the convex clustering problem. This algorithm belongs to a subclass of MM (minimization-majorization) algorithms known as proximal distance algorithms. The proximal distance convex clustering algorithm is inherently parallelizable and readily maps to modern many-core devices such as graphics processing units (GPUs). Our freely available software, CONVEXCLUSTER, exploits OpenCL routines that ensure compatibility across a variety of hardware environments.

This is a *PLOS Computational Biology* Methods paper.

Introduction

Pattern discovery is one of the primary goals of bioinformatics. Cluster analysis is a broad term for a variety of exploratory methods that reveal patterns based on similarities between data points. Well-known methods such as *k*-means invoke a fixed number of clusters. In complex biological data, the number of clusters is unknown in advance, and it is appealing to vary the number of clusters simultaneously with cluster assignment. Hierarchical clustering has been particularly helpful in understanding cluster granularity in gene-expression studies and other applications. In addition to producing easily visualized and interpretable results, hierarchical clustering is simple to implement and computationally quick. These are legitimate advantages, but they do not compensate for hierarchical clustering's instability to small data perturbations such as measurement error. Cluster inference can be adversely affected as small errors accumulate.

All principled methods of clustering attempt to minimize some measure of within group dissimilarity. Hierarchical clustering constructs a bifurcating tree by fusing or dividing observations (features). Fusion is referred to as agglomerative clustering and splitting as divisive clustering. Because of the greedy nature of the choices in hierarchical clustering, it returns clusters that are only locally optimal with respect to the underlying criterion [1]. Solution quality may vary depending on how clusters are fused. There is no guarantee that UPGMA, single linkage, or complete linkage will agree or will collectively or individually give the optimal clusters. A potentially greater handicap is that small perturbations in the data can lead to large changes in hierarchical clustering assignments. This propensity makes hierarchical clustering sensitive to outliers and promotes the formation of spurious clusters. In combination, the presence of local minima and the sensitivity to outliers lead to irreproducible results.

Although hierarchical clustering has its drawbacks, completely reformulating it might be detrimental. Recently [2] and [3] introduced convex clustering based on minimizing a penalized sum of squares. Their criterion is coercive and strictly convex. Recall that a function $f(\boldsymbol{x})$ is coercive if $\lim_{\|\boldsymbol{x}\| \to \infty} f(\boldsymbol{x}) = \infty$. According to a classical theorem of mathematical analysis, a continuous coercive function achieves its minimum. Strict convexity of the convex clustering criterion ensures that the global minimizer is unique. The penalty term in convex clustering criterion accommodates prior information through nonuniform weights on data pairs. The solution paths of convex clustering retain the straightforward interpretability of hierarchical clustering while ameliorating its sensitivity to outliers and tendency to get trapped by local minima.

Despite the persuasive advantages of convex clustering, there are two obstacles that stand in its way of becoming a practical tool in bioinformatics. The first is the challenge of large-scale problems. Current algorithms are computationally intensive and scale poorly on high-dimensional problems. A second obstacle is the minimal guidance currently available on how to choose penalty weights. Hocking and colleagues suggest some rules of thumb but offer little detailed advice [3]. In our experience, the quality of the clustering path depends critically on well-designed weights. To address these issues, the current paper describes a fast new algorithm and a corresponding software implementation, CONVEXCLUSTER. Our advice on strategies for choosing penalty weights is grounded in some practical biological examples. These examples support our conviction that convex clustering can be more nuanced than hierarchical clustering. Our examples include Fisher's Iris data from discriminant analysis, ethnicity clustering based on microsatellite genotypes from the Human Genome Diversity Project and SNP genotypes from the POPRES project, and breast cancer subtype classification via microarrays. In the POPRES data, we first reduce the genotypes to principal components and then use these to cluster. The paths computed under convex clustering expose features of the data hidden to less sophisticated clustering methods. The potential for understanding human evolution and history alone justify wider adoption of convex clustering.

Methods

Assume that there are n cases and p features. For example, cases might correspond to cancer patients and features to their biomarker profiles. The more vivid language of graph theory speaks of nodes rather than cases and edges rather than pairs of cases. To implement convex clustering, [2] suggest minimizing the penalized loss function

$$f_{\mu}(\boldsymbol{U}) \quad = \quad \frac{1}{2}\sum_{i=1}^{n}||\boldsymbol{x}_i - \boldsymbol{u}_i||^2 + \mu\sum_{i<j} w_{ij}||\boldsymbol{u}_i - \boldsymbol{u}_j|| \tag{1}$$

relying on Euclidean norms. Here the column vector $\boldsymbol{x}_i \in \mathbb{R}^p$ of the matrix $\boldsymbol{X} \in \mathbb{R}^{p \times n}$ records the features for case i, the column $\boldsymbol{u}_i$ of the matrix $\boldsymbol{U}$ denotes the cluster center assigned to case i, $\mu \geq 0$ tunes the strength of the penalty, and $w_{ij} \geq 0$ weights the contribution of the case pair (i, j) to the penalty. Unless sparse, the weights w_{ij} are stored in a symmetric $n \times n$ adjacency matrix. Fig 1 illustrates the concept of convex clustering on three data point extracted from the Iris dataset [4]. The objective function $f_{\mu}(\boldsymbol{U})$ treats the features symmetrically. If these range over widely varying scales, it is prudent to standardize each feature to have mean 0 and variance 1.

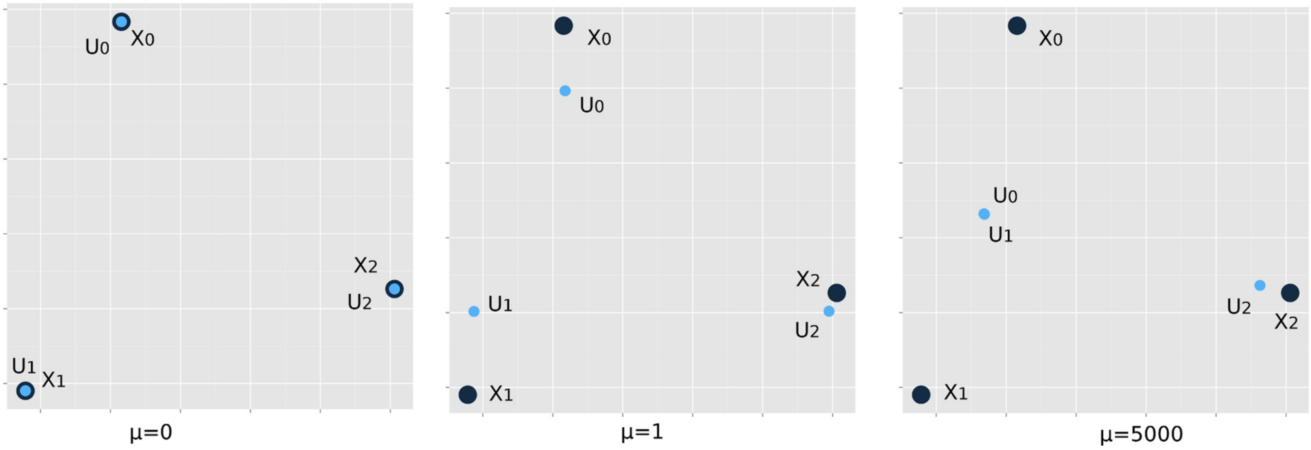

Fig 1. Convex clustering concepts. For clarity, we present three random data points extracted from the three classes in the Iris dataset. Black points denote the original data points ***X*** and blue points denote the cluster centers ***U***. At $\mu = 0$, ***X*** and ***U*** coincide. At intermediate μ values (middle figure), ***U*** coalesces towards its cluster center. For sufficiently large μ, ***U*** converges to cluster centers (right figure). Note that in this demonstration, only the left two points have non-zero pairwise weights w_{ij}. Hence, the two resulting clusters reflect the two graphs defined by the matrix of weights.

doi:10.1371/journal.pcbi.1004228.g001

Because the objective function $f_\mu(\boldsymbol{U})$ is strictly convex and coercive, a unique minimum point exists for each value of μ. When $\mu = 0$ and the $\boldsymbol{x}_i$ are unique, the choices $\boldsymbol{u}_i = \boldsymbol{x}_i$ minimize $f_\mu(\boldsymbol{U})$, and there are as many clusters as cases. If the underlying graph is connected, then as μ increases, cluster centers coalesce until all centers merge into a single cluster with all $\boldsymbol{u}_i = \bar{\boldsymbol{x}}$, the average of the data points $\boldsymbol{x}_i$. Although splitting events as well as fusion events can in principle occur along the solution path, following the path as μ increases typically reveals a hierarchical structure among the clusters. The weights encode prior information that guides clustering. Setting some of the weights equal to 0 reduces the computational load of minimizing $f_\mu(\boldsymbol{U})$ in the proximal distance algorithm introduced next.

The Proximal Distance Algorithm

The proximal distance principle is a new way of attacking constrained optimization problems [5]. The principle is capable of enforcing parsimony in parameter estimation while avoiding the shrinkage incurred by convex penalties such as the lasso. In parametric models, shrinkage leads to biased parameter estimates and entices false positives to enter the model. Imperfect models in turn fit new data poorly. The proximal distance principle seeks to minimize a function $h(\boldsymbol{y})$, possibly nonsmooth, subject to $\boldsymbol{y} \in C$, where C is a closed set, not necessarily convex. The set C encodes constraints such as sparsity. In the exact penalty method of Clarke [6, 7, 8], this constrained problem is replaced by the unconstrained problem of minimizing $h(\boldsymbol{y}) + \rho \operatorname{dist}(\boldsymbol{y}, C)$, where $\operatorname{dist}(\boldsymbol{y}, C)$ denotes the Euclidean distance from $\boldsymbol{y}$ to C. Note that $\operatorname{dist}(\boldsymbol{y}, C) = 0$ is a necessary and sufficient condition for $\boldsymbol{y} \in C$. If ρ is chosen large enough, say bigger than a Lipschitz constant for $h(\boldsymbol{y})$, then the minima of the two problems coincide (Proposition 6.3.2 in [6]).

How does convex clustering fit in this abstract framework? Although the objective function $f_\mu(\boldsymbol{U})$ is certainly nonsmooth, there are no constraints in sight. The strategy of parameter splitting introduces constraints to simplify the objective function. Since least squares problems are routine, the penalty terms constitute the intractable part of the objective function $f_\mu(\boldsymbol{U})$. One can simplify the term $\|\boldsymbol{u}_i - \boldsymbol{u}_j\|$ by replacing the vector difference $\boldsymbol{u}_i - \boldsymbol{u}_j$ by the single vector $\boldsymbol{v}_{ij}$

and imposing the constraint $\boldsymbol{v}_{ij} = \boldsymbol{u}_i - \boldsymbol{u}_j$. Parameter splitting therefore leads to the revised objective function

$$g_\mu(\boldsymbol{U}, \boldsymbol{V}) \quad = \quad \frac{1}{2}\sum_{i=1}^{n} ||\boldsymbol{x}_i - \boldsymbol{u}_i||^2 + \mu \sum_{i<j} w_{ij} ||\boldsymbol{v}_{ij}|| \tag{2}$$

with a simpler loss, an expanded set of parameters, and a linear constraint set C encapsulating the pairwise constraints $\boldsymbol{v}_{ij} = \boldsymbol{u}_i - \boldsymbol{u}_j$.

The proximal distance method undertakes minimization of $h(\boldsymbol{y}) + \rho\,\mathrm{dist}(\boldsymbol{y}, C)$ by a combination of approximation, the MM (majorization-minimization) principle [9, 10, 11, 12, 13], and an appeal to a combination of set projection [14] and proximal mapping [15]. The latter operations have been intensely studied for years and implemented in a host of special cases. Thus, the proximal distance principle encourages highly modular solutions to difficult optimization problems. Furthermore, most proximal distance algorithms benefit from parallelization.

Let us consider each of the ingredients of the proximal distance algorithm in turn, starting with approximation. The function $\mathrm{dist}(\boldsymbol{y}, C)$ is nonsmooth even when C is well behaved. For $\epsilon > 0$ small, the revised distance $\mathrm{dist}_\epsilon(\boldsymbol{y}, C) = \sqrt{(\mathrm{dist}(\boldsymbol{y}, C))^2 + \epsilon}$ is differentiable and approximates $\mathrm{dist}(\boldsymbol{y}, C)$ well. The MM principle leads to algorithms that systematically decrease the objective function. In the case of minimizing $f(\boldsymbol{y}) + \rho\,\mathrm{dist}(\boldsymbol{y}, C)$ one can invoke the majorization $\mathrm{dist}(\boldsymbol{y}, C) \leq \|\boldsymbol{y} - P_C(\boldsymbol{y}_m)\|$, where $P_C(\boldsymbol{y}_m)$ is the projection of the current iterate $\boldsymbol{y}_m$ onto the set C. By definition $\mathrm{dist}(\boldsymbol{y}_m, C) = \|\boldsymbol{y}_m - P_C(\boldsymbol{y}_m)\|$, and $P_C(\boldsymbol{y}_m)$ is a closest point in C to the point $\boldsymbol{y}_m$. For a closed nonconvex set, there may be multiple closest points; for a closed convex set there is exactly one.

According to the MM principle, minimizing the surrogate function

$$\frac{1}{2}\sum_{i=1}^{n} ||\boldsymbol{x}_i - \boldsymbol{u}_i||^2 + \mu \sum_{i<j} w_{ij} ||\boldsymbol{v}_{ij}|| + \rho \sqrt{\left\| \begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{V} \end{pmatrix} - P_C \begin{pmatrix} \boldsymbol{U}_m \\ \boldsymbol{V}_m \end{pmatrix} \right\|^2 + \epsilon} \tag{3}$$

drives the approximate objective function

$$\frac{1}{2}\sum_{i=1}^{n} ||\boldsymbol{x}_i - \boldsymbol{u}_i||^2 + \mu \sum_{i<j} w_{ij} ||\boldsymbol{v}_{ij}|| + \rho \sqrt{\mathrm{dist}\left[\begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{V} \end{pmatrix}, C \right]^2 + \epsilon}$$

downhill. The surrogate function Eq (3) is still too complicated for our purposes. The remedy is another round of majorization. This time the majorization

$$\sqrt{t + \epsilon} \quad \leq \quad \sqrt{t_m + \epsilon} + \frac{1}{2\sqrt{t_m + \epsilon}} (t - t_m) \tag{4}$$

comes into play based on the concavity of the function $\sqrt{t + \epsilon}$ for $t \geq 0$. This follows from the fact that a differentiable concave function is always bounded by its first order Taylor expansion. As required by the MM principle, equality holds in the majorization Eq (4) when $t = t_m$.

Applying this majorization to the surrogate function Eq (3) yields the new surrogate

$$\begin{aligned} h[(\boldsymbol{U},\boldsymbol{V})|(\boldsymbol{U}_m,\boldsymbol{V}_m)] &= \frac{1}{2}\sum_{i=1}^{n}||\boldsymbol{x}_i-\boldsymbol{u}_i||^2+\mu\sum_{i<j}w_{ij}||\boldsymbol{v}_{ij}||+\frac{\rho}{2d_m}\left|\left|\begin{pmatrix}\boldsymbol{U}\\ \boldsymbol{V}\end{pmatrix}-P_C\begin{pmatrix}\boldsymbol{U}_m\\ \boldsymbol{V}_m\end{pmatrix}\right|\right|^2 \\ d_m &= \sqrt{\left|\left|\begin{pmatrix}\boldsymbol{U}_m\\ \boldsymbol{V}_m\end{pmatrix}-P_C\begin{pmatrix}\boldsymbol{U}_m\\ \boldsymbol{V}_m\end{pmatrix}\right|\right|^2+\epsilon} \end{aligned} \tag{5}$$

up to an irrelevant constant. The surrogate function Eq (5) resulting from these maneuvers separates all of the vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_{ij}$. The derivative of the surrogate with respect to $\boldsymbol{u}_i$ is

$$\frac{\partial}{\partial\boldsymbol{u}_i}h[(\boldsymbol{U},\boldsymbol{V})|(\boldsymbol{U}_m,\boldsymbol{V}_m)] = \boldsymbol{u}_i-\boldsymbol{x}_i+\frac{\rho}{d_m}(\boldsymbol{u}_i-\boldsymbol{a}_{n,i}),$$

where $\boldsymbol{a}_{n,i}$ is the part of the projection pertaining to $\boldsymbol{u}_i$. One can explicitly solve for the update

$$\boldsymbol{u}_{n+1,i} = \frac{d_m}{d_m+\rho}\boldsymbol{x}_i+\frac{\rho}{d_m+\rho}\boldsymbol{a}_{n,i}.$$

The update of $\boldsymbol{v}_{ij}$ involves shrinkage. Let $\boldsymbol{b}_{n,ij}$ denote the part of the projection pertaining to $\boldsymbol{v}_{ij}$. Standard arguments from convex calculus [16] show that the minimum of $\mu w_{ij}||\boldsymbol{v}_{ij}||+\frac{\rho}{2d_m}||\boldsymbol{v}_{ij}-\boldsymbol{b}_{n,ij}||^2$ is achieved by

$$\boldsymbol{v}_{n+1,ij} = \max\left\{\left(1-\frac{\mu w_{ij}d_m}{\rho\parallel\boldsymbol{b}_{n,ij}\parallel}\right),0\right\}\boldsymbol{b}_{n,ij}. \tag{6}$$

In the exceptional case $\boldsymbol{b}_{n,ij}=\boldsymbol{0}$, the solution $\boldsymbol{v}_{n+1,ij}=\boldsymbol{0}$ is clear from inspection of the $\boldsymbol{v}_{ij}$ criterion Eq (6). Both of these solution maps fall under the heading of proximal operators, hence, the name proximal distance algorithm.

If a weight $w_{ij}=0$, then it is computationally inefficient to introduce a difference vector $\boldsymbol{v}_{ij}$. Thus, in many applications, the weight matrix $\boldsymbol{W}=(w_{ij})$ may be sparse. The block descent algorithm for projection, that we discuss next, takes into account the sparsity patterns in $\boldsymbol{W}$. Again taking the sparsity pattern of $\boldsymbol{W}$ into account enables us to employ fewer difference vectors. Let E denote the set of edges $\{i,j\}$ with positive weights $w_{ij}=w_{ji}$. Divide the neighborhood N_i of a node i into left and right node neighborhoods $L_i=\{j<i{:}w_{ji}>0\}$ and $R_i=\{j>i{:}w_{ij}>0\}$. Clearly $N_i=L_i\cup R_i$, and $E=\cup_{i=1}^{n}N_i$. Projection minimizes the criterion

$$\frac{1}{2}\sum_{i=1}^{n}\parallel\boldsymbol{u}_i-\tilde{\boldsymbol{u}}_i\parallel^2+\frac{1}{2}\sum_{\{i,j\}\in E}\parallel\boldsymbol{u}_i-\boldsymbol{u}_j-\tilde{\boldsymbol{v}}_{ij}\parallel^2$$

for $\tilde{\boldsymbol{U}}$ and $\tilde{\boldsymbol{V}}$ given. One can minimize this criterion by equating its derivative with respect to $\boldsymbol{u}_i$ to $\boldsymbol{0}$. It is unclear how to massage the stationarity equation

$$\boldsymbol{0} = \boldsymbol{u}_i-\tilde{\boldsymbol{u}}_i+\sum_{j\in R_i}(\boldsymbol{u}_i-\boldsymbol{u}_j-\tilde{\boldsymbol{v}}_{ij})-\sum_{j\in L_i}(\boldsymbol{u}_j-\boldsymbol{u}_i-\tilde{\boldsymbol{v}}_{ji})$$

into a solvable form. However, the block updates

$$\boldsymbol{u}_i = \frac{1}{1+|N_i|}\left(\tilde{\boldsymbol{u}}_i+\sum_{j\in R_i}\tilde{\boldsymbol{v}}_{ij}-\sum_{j\in L_i}\tilde{\boldsymbol{v}}_{ji}+\sum_{j\in N_i}\boldsymbol{u}_j\right)$$

are available. Here $|N_i|$ denotes the cardinality of N_i. One cycle of the block descent algorithm updates $\boldsymbol{u}_1$ through $\boldsymbol{u}_n$ sequentially. This cycle is repeated until all of the vectors $\boldsymbol{u}_i$ stabilize. Once convergence is achieved, one sets $\boldsymbol{v}_{ij} = \boldsymbol{u}_i - \boldsymbol{u}_j$ for the relevant pairs.

Missing Data

In general, clustering methods require complete data. The remedy of pre-imputation of missing values can be sensitive to the model assumptions underlying a given imputation method. A better remedy is to change the clustering criterion to directly reflect missing data. It is then straightforward to accommodate missing data in $\boldsymbol{X}$ by another round of majorization. Suppose Γ is the set of ordered index pairs (i, j) corresponding to the observed entries x_{ij} of $\boldsymbol{X}$. We now minimize the revised criterion

$$f_\mu(\boldsymbol{U}) \quad = \quad \frac{1}{2}\sum_{(i,j)\in\Gamma}(x_{ij} - u_{ij})^2 + \mu\sum_{i<j} w_{ij}\|\boldsymbol{u}_i - \boldsymbol{u}_j\|, \tag{7}$$

which unfortunately lacks the symmetry of the original problem. To restore the lost symmetry, we invoke the majorization

$$\frac{1}{2}\sum_{(i,j)\in\Gamma}(x_{ij} - u_{ij})^2 \quad \leq \quad \frac{1}{2}\sum_{(i,j)\in\Gamma}(x_{ij} - u_{ij})^2 + \frac{1}{2}\sum_{(i,j)\notin\Gamma}(u_{mij} - u_{ij})^2,$$

where u_{mij} is a component of $\boldsymbol{U}_m$. In essence, the term $(u_{mij} - u_{ij})^2$ majorizes 0. If the $n \times p$ matrix $\boldsymbol{Y} = (y_{ij})$ has entries $y_{ij} = x_{ij}$ for $(i, j) \in \Gamma$ and $y_{ij} = u_{mij}$ for $(i, j) \notin \Gamma$, then in the minimization step of the proximal distance algorithm, we simply minimize the surrogate function

$$g_\mu(\boldsymbol{U}, \boldsymbol{V}) \quad = \quad \frac{1}{2}\sum_{i=1}^{n}\|\boldsymbol{y}_i - \boldsymbol{u}_i\|^2 + \mu\sum_{i<j} w_{ij}\|\boldsymbol{v}_{ij}\| \tag{8}$$

The rest of the proximal distance algorithm remains the same.

Calibration of Weights

The pairwise weight $w_{ij} = w_{ji}$ introduced in the penalty term of Eq (1) determines the importance of similarity between nodes i and j. Two principles guide our choice of weights. First, the weight w_{ij} should be inversely proportional to the distance between the ith and jth points. This inverse relationship accords with intuition. As w_{ij} increases, the pressure for the ith and jth centroids to coalesce increases. If the weights w_{ij} are correlated with the similarity of the feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, then the pressure for their centroids to merge is especially great. Second, the weight matrix $\boldsymbol{W}$ should be sparse. Despite the fact that small positive weights and zero weights lead to similar clustering paths, the computational advantages of zero weights cannot be ignored.

These observations prompt the following choice of weights. To maintain computational efficiency, it is helpful to focus on the k nearest neighbors of each node. We define the distance d_{ij} between two nodes i and j by the Euclidean norm $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and write $i \sim_k j$ if j occurs among the k nearest neighbors of i or i occurs among the k nearest neighbors of j. Based on these considerations the weights

$$w_{ij} \quad = \quad 1_{\{i\sim_k j\}} e^{-\phi d_{ij}^2} \tag{9}$$

are reasonable, where $1_{\{i \sim_k j\}}$ is the indicator function of the event $\{i \sim_k j\}$ and $\phi \geq 0$ is a tuning constant. The case $\phi = 0$ corresponds to uniform weights between nearest neighbors. When ϕ is positive, w_{ij} strictly decreases as a function of d_{ij}. Complete coalescence of the nodes occurs as μ increases if the graph is connected based on all w_{ij}. Using squared distances d_{ij}^2 rather than distances d_{ij} induces more aggressive coalescence of nearby points and slower coalescence of distant points. In practice we normalize weights so that they sum to 1. This harmless tactic is equivalent to rescaling μ. This generic framework was proposed by [3].

We now discuss a strategy for leveraging additional information. When expert knowledge on the relationships among nodes is available and can be quantified, incorporating such knowledge may improve the clustering path. This must be done delicately so that prior information does not overwhelm observed data. If $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ store the genotypes and GPS (global positioning system) coordinates of subject i, respectively, then the weighted average

$$d_{ij} \quad = \quad \alpha \parallel \boldsymbol{x}_i - \boldsymbol{x}_j \parallel + (1-\alpha) \parallel \boldsymbol{y}_i - \boldsymbol{y}_j \parallel, \qquad \alpha \in (0,1), \tag{10}$$

serves as a composite distance helpful in clustering subjects. In Eq (10) observe that the components of the difference $\boldsymbol{y}_i–\boldsymbol{y}_j$ must be computed in modulo arithmetic. Given a proper choice of the scaling constant α, an even better alternative replaces $\|\boldsymbol{y}_i–\boldsymbol{y}_j\|$ by the geodesic distance between i and j. One could reverse the roles of the vector pairs $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$, but it seems to us that genotype similarity rather than physical proximity should be the primary driver of clustering. GPS coordinates are less informative, crudely estimated, and shared across many cases.

Evaluation of Clusters

Our program CONVEXCLUSTER minimizes the penalized loss Eq (2) for a range of user specified μ values. For each μ the optimized matrix $\boldsymbol{U}$ of cluster centers is stored in a temporary file for later construction of the cluster path. To facilitate visualization, CONVEXCLUSTER encourages users to project the cluster path onto any two principal components of the original data. The first example of Section 1 relies on the classical Iris data of discriminant analysis [4]. This dataset contains 150 cases spread over three species. The Iris data can be downloaded from the UCI machine learning repository [17]. For purposes of comparison, we also evaluated the clusters formed by agglomerative hierarchical clustering. In contrast to convex clustering, hierarchical clustering results are usually visualized via dendrograms. Hierarchical clustering comes in several flavors; we chose UPGMA (Unweighted Pair Group Method with Arithmetic Mean) [18] as implemented in the R function *hclust*. Although *hclust* offers six other options for merging clusters, UPGMA is probably the most reliable in reducing the detrimental effects of outliers since it averages information across all cluster members. UPGMA operates on a matrix of pairwise distances defined between nodes. In our genetics examples, we take these to be the distances defined by Eq (10). To make a fair comparison between convex and hierarchical clustering, we invoke the composite distance in both methods. We also present results graphically by projecting cluster paths onto the first two principal components of the genetic data in Examples 1 and 2 and the expression data in the last Example. To generate a cluster path for hierarchical clustering, we assigned each fusion node on the tree as as the average of the values of its descendant leaves.

Results

Guidance on Selecting Constants k and ϕ

In computing pairwise weights, one is immediately confronted with the question of how to select the constants k (number of nearest neighbors) and ϕ (the soft-threshold effect). The answer depends upon one's research goals. Unlike supervised learning such as classification, clustering is inherently exploratory. In practice it usually looks for coarse-level relationships among the data points before drilling down in coarse clusters to look for fine-level relationships. In

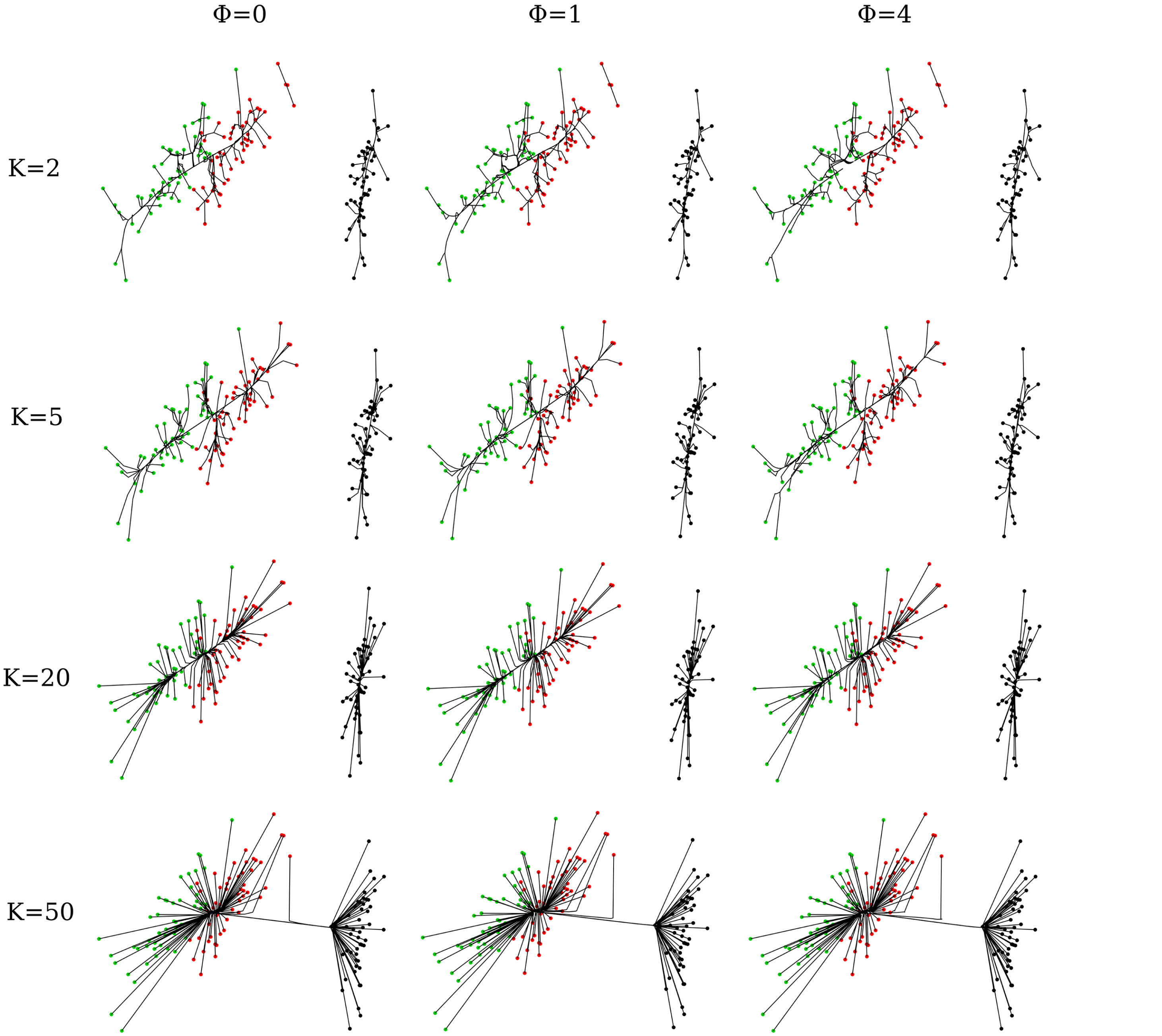

Fig 2. Effects of the parameters *k* and ϕ on cluster paths in the Iris data. Black, red, and green points denote the species Iris-setosa, Iris-versicolor, and Iris-virginica, respectively. These points are projections of the Iris dataset on the first two principal components (PCs). Lines trace the cluster centers as they traverse the regularization path. The subtle impact of ϕ is revealed in two cases. At $k = 50$, a red dot coalesces with the right cluster at $\phi = 0$, but with the left cluster for larger values of ϕ. At $k = 5$ or $k = 10$, the two green dots at the extreme lower left corner coalesce later at the largest value of ϕ.

doi:10.1371/journal.pcbi.1004228.g002

hierarchical clustering different levels of granularity can be explored by drawing a line bisecting all branches along a given level of the tree. Our recommendation for convex clustering is to begin with large values of k and then examine the patterns revealed as k is progressively reduced. All points eventually coalesce to a single cluster while k exceeds a particular threshold, which is determined by the separation of the nodes.

To get a sense of the impact of the constants k and ϕ on the Iris data, we generated cluster paths for various pairs (k, ϕ). As Fig 2 illustrates, k quantifies the connectivity of the underlying graph. Eventual coalescence only occurs for $k = 50$; even then the apparent Iris-Versicolor outlier does not coalesce until very late. All values of k support a clear separation of Iris-Setosa from the other two species Iris-Versicolor and Iris-Virginica. Separation of Iris-Versicolor and Iris-Virginica into two different groups becomes discernible at $k = 20$. Subgroups within each species are evident for $k = 5$ and $k = 2$. Improved resolution comes at a price; the two small two-member clusters seen in the top right corner of the main Iris-Versicolor cluster never fully coalesce with the main cluster when $k = 2$. The distance tuning constant ϕ also exerts a subtle influence along each row of Fig 2. This influence is more strongly felt for low values of k. For example, for $k = 2$ and $k = 5$ with $\phi = 4$, we observe that the two green points at the bottom left of the cluster graph coalesce much later when ϕ is set to smaller values. Examination of the Iris data suggests exploring cluster granularity over a range of k values with ϕ set to 0. One can find the minimum k ensuring full connectivity by combining bisection* with either breadth-first search or depth-first search [19]. Once the desired granularity is achieved, ϕ can be increased to reveal more subtle details. Note that increasing ϕ sends most weights between k nearest neighbors to 0. As previously noted, the proximal distance algorithm takes substantially more iterations to converge for large values of ϕ.

As the Iris data illustrate, cluster inference is robust over a wide range of k values. Across all four rows in Fig 2, we would have learned that there are two major classes of Iris, even if the points were plotted in the same color. By decreasing k, we were able to discern relationships within the two classes. The figure also shows that the parameter ϕ is less critical than k. Note, however, that for low values of k, better resolution is achieved by increasing ϕ from 0.

Cluster Accuracy in the Presence of Noise

Although agglomerative hierarchical clustering is computationally efficient, it is greedy, and greedy algorithms tend to produce suboptimal solutions [1]. In particular, it can falter in the face of noisy data. To test this hypothesis, we simulated new data from the Iris data. In creating a dataset, we perturbed each row of the data matrix $\boldsymbol{X}$ by adding normal deviates with mean 0

Table 1. Avg Rand indices (RI) as a function of noise in the Iris data.

Noise level *c*	HCLUST	CONVEXCLUSTER		
	UPGMA RI	k = 5 RI	k = 10 RI	k = 15 RI
0.02	.83(.05)	.88(.03)	.89(.01)	.89(.02)
0.04	.83(.05)	.88(.03)	.88(.02)	.88(.03)
0.06	.83(.05)	.88(.03)	.88(.03)	.88(.03)
0.08	.82(.05)	.88(.04)	.88(.03)	.87(.03)
0.10	.82(.05)	.87(.04)	.87(.04)	.86(.04)

Standard deviations in parentheses. For computational efficiency, ϕ was set to zero for convex clustering.

doi:10.1371/journal.pcbi.1004228.t001

Table 2. Avg Rand indices (RI) as a function of missingness in the Iris data.

Proportion of rows with a missing attribute *c*	HCLUST	CONVEXCLUSTER		
	UPGMA RI	k = 5 RI	k = 10 RI	k = 15 RI
0.25	.82(.05)	.88(.03)	.88(.03)	.87(.02)
0.50	.83(.05)	.87(.04)	.86(.03)	.86(.03)
0.75	.82(.05)	.86(.05)	.85(.04)	.86(.04)
1.00	.82(.04)	.86(.05)	.84(.05)	.85(.04)

Standard deviations in parentheses. For computational efficiency, ϕ was set to zero for convex clustering.

doi:10.1371/journal.pcbi.1004228.t002

and standard deviation equal to the sample standard deviation s^2 of the corresponding feature multiplied by a constant *c*. We then clustered the data points into three clusters and quantitatively evaluated clustering performance through Normalized Rand Indices [20]. For convex clustering, visual inspection of the converged clustering paths reveals roughly three major clusters for values of *k* between 5 and 15. With hierarchical clustering, three clusters were constructed by choosing a cut point on the full tree intersecting three branches. Table 1 summarizes Rand indices averaged over 100 replicates under the two methods. Larger values of the Rand index represent higher accuracy; the maximum value of 1 indicates error-free clustering. Examination of the table suggests that convex clustering is indeed more accurate in the face of noise over a wide range of *k* values.

Cluster Accuracy with Missing Values

We carried out a second simulation study on the Iris data to assess accuracy of cluster inference as a function of missingness. Because the Iris data includes only four features (width and height of sepals and petals), simply selecting entries of the data matrix at random can lead to cases retaining no data. To avoid these degeneracies, we randomly selected cases and then a random feature from each case for deletion. Given cases rates of 25%, 50%, 75%, and 100%, the proportion of missing observations consequently ranged from 5% to 25%. Hierarchical clustering with missing data requires that either cases with missing entries be omitted or that missing entries be imputed. We employed the second strategy, filling in missing entries by multiple imputation as implemented in the R package MI [21]. Hierarchical clustering was then applied to the completed data. For convex clustering, we also applied multiple imputation, but for the sole purpose of computing the convex clustering weights. We then applied convex clustering to the original incomplete data under the objective function Eq (7). Accuracy for each method was estimated in the same manner as the previous simulations. The Rand indices in Table 2 suggest that convex clustering does indeed outperform hierarchical clustering in the presence of missing data.

Inference of Ethnicity

As genotyping costs have dropped in recent years, it has become straightforward to relate ethnicity to subtle genetic variations. Several software tools are now available for this purpose. For example, the programs STRUCTURE [22] and ADMIXTURE [23] estimate a subject's admixture proportions across a set of predefined or inferred ancestral populations. EIGENSTRAT [24] employs a handful of principal components to explain ethnic variation. Principal component analysis (PCA) is attractive due to its speed and ease of visualization. Clustering can also separate subjects by ethnicity if individuals of mixed ethnicity are omitted. The advantage of convex clustering is that one can follow the dynamic behavior of the relationship clusters along the regularization path. In the next two examples on population structure, the data consist of multidimensional genotypes. We project our convex clustering paths onto the first two principal components of the data. This produces plots where population substructure aligns with geographic regions of origin.

World-wide genetic diversity. For a practical demonstration of convex clustering, we now turn to the Human Genome Diversity Project (HGDP). This collaboration makes several datasets publicly available that vary in marker type (SNPs versus microsatellites) and sample size. The HGDP 2002 dataset considered here includes 1,056 individuals from 52 populations genotyped at 377 autosomal microsatellites [25]. Care must be taken in analyzing microsatellites since, in contrast to SNPs, they display more alleles and greater levels of polymorphism. Recall that an allele at a microsatellite approximates the number of short tandem repeats of some simple motif. Because treating microsatellite genotypes as continuous variables is problematic, we encode each microsatellite genotype as a sequence of allele counts. Each count ranges from 0 to 2, and there are as many count variables as alleles. This encoding yields a revised 2002 dataset with the 377 microsatellite genotypes expanded to 4,682 different attributes.

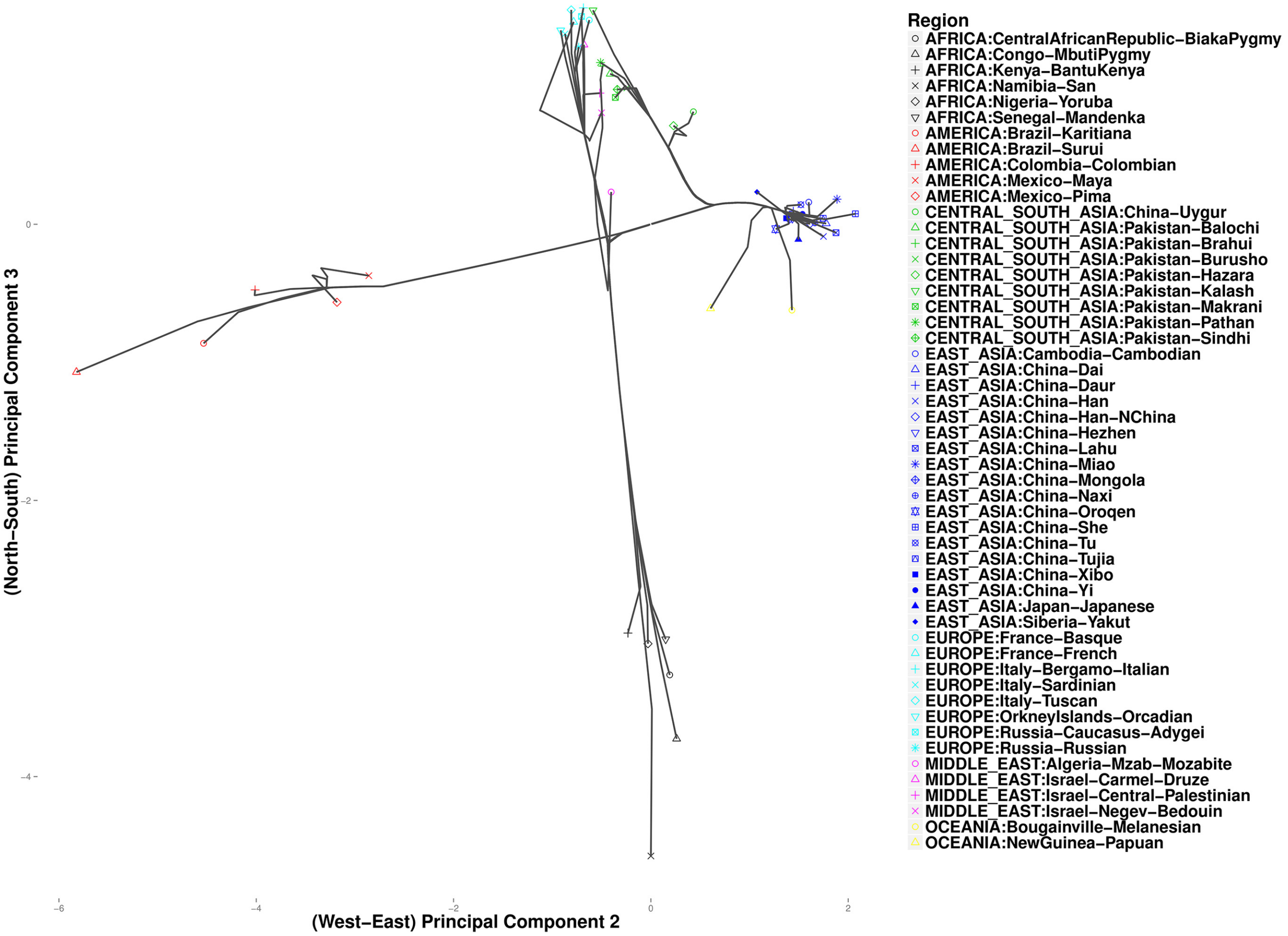

Fig 3. Convex clustering of the HGDP data using a large number of nearest neighbors to infer intercontinental connections ($k = 4$, $\phi = 1$).

doi:10.1371/journal.pcbi.1004228.g003

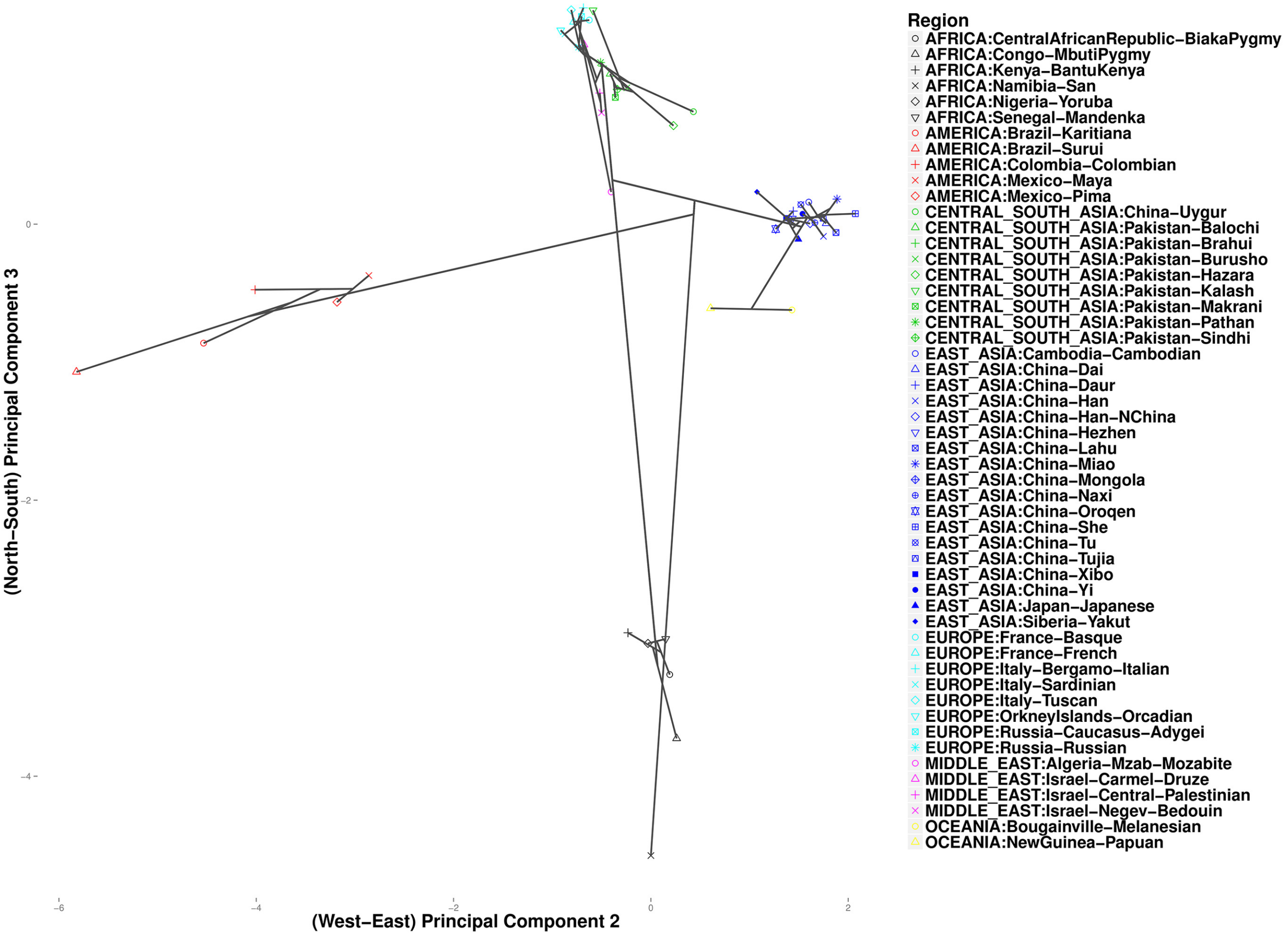

Fig 4. Hierarchical clustering of the 52 populations from the HGDP data.

doi:10.1371/journal.pcbi.1004228.g004

As expected, these data exhibit clines in allele frequencies [26]. To take advantage of the correlation between geographic separation and ethnic similarity, we defined penalty weights w_{ij} according to the composite distance in Eq (10) with constant $\alpha = 0.5$. We chose this value of α to give equal weight to both sources of information. Results for other values of $\alpha \in (0, 1)$ are similar. We progressively reduced k from a large value such as 10 until we could observe separation of the seven major continental groups. Variations in ϕ make no discernible differences in the analysis of these data. Fig 3 plots cluster paths for these data given the settings $\phi = 1$ and $k = 4$. With $k = 4$ nearest neighbors, we observe broad-scale clustering events that link up the major continental groups. In the north, Europeans fall into a single cluster, later joined by populations from the Middle East. In the east the Chinese merge into a cluster that subsequently merges with two Oceania populations from New Guinea. This mega cluster then merges with various Central Asian populations of predominantly Pakistani origin. In the west five Central/South American populations cluster, and in the south six African populations cluster. Considering the continental clusters in the figure, the American cluster (red points) and the Central/East Asian cluster (green points) are linked by a straight line, while the northern (turquoise, green, and magenta points) and southern continental clusters (black points) appear to fuse at a

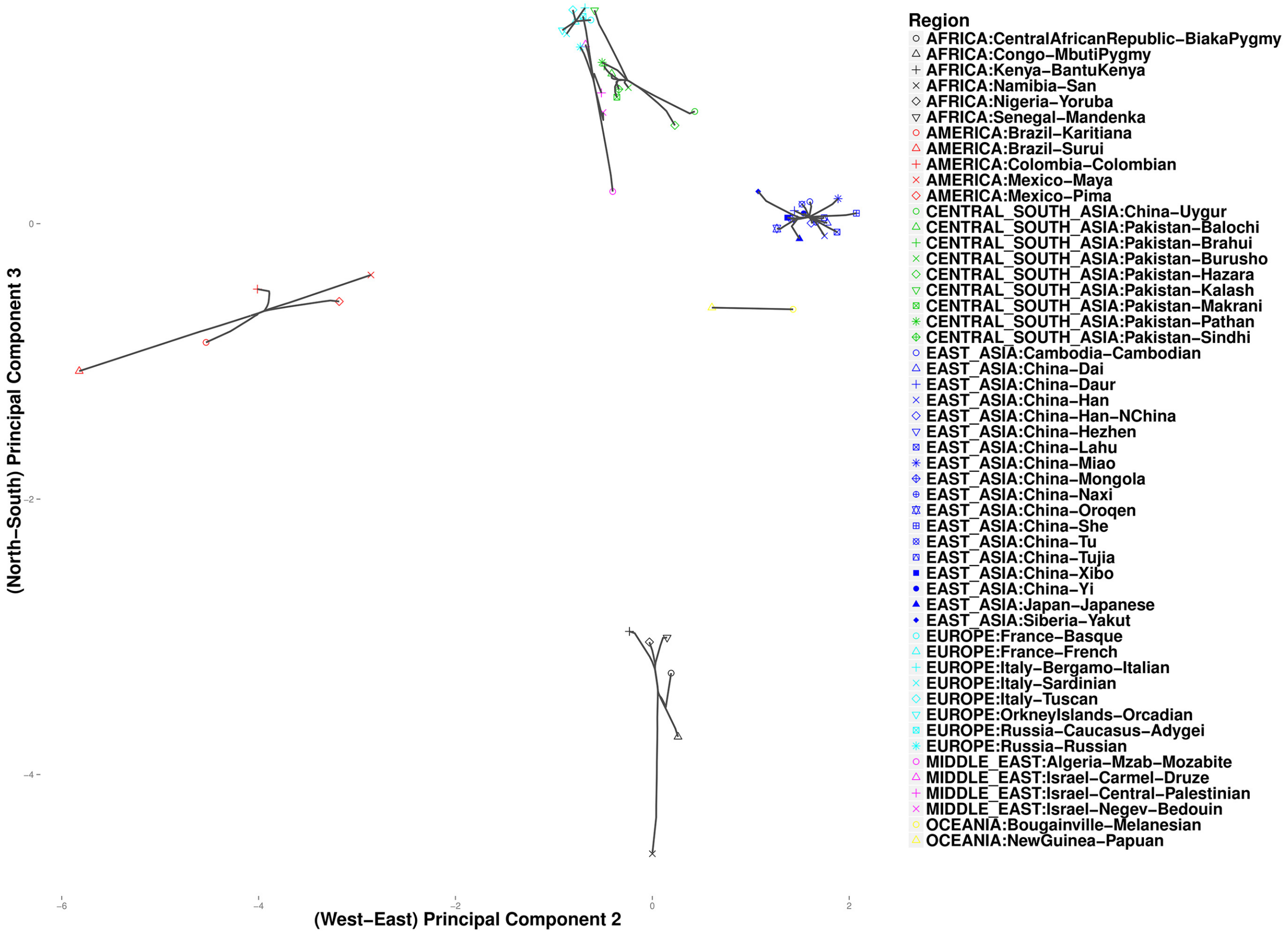

Fig 5. Convex clustering of the HGDP data using a small number *k* of nearest neighbors to resolve intracontinental connections ($k = 1$, $\phi = 1$).

doi:10.1371/journal.pcbi.1004228.g005

point just below this straight line. This accords with known links between East Asians and American Indians, who crossed the Bering strait, possibly multiple times, during the Ice Age [27]. Fig 4 presents the output of hierarchical clustering, where datapoints and their fused (averaged) values are projected onto the same coordinates as the convex clustering results. Although the two methods give fairly consistent plots, there is a striking difference in how the African San population is treated. In hierarchical clustering it coalesces to the origin as a single outlier continental region. The Central Asian groups also appear to be more closely related to Europeans. In convex clustering Fig 5 depicts finer grained events exposed by setting $k = 1$. Along the western axis, taking $k = 1$ is uninformative, but among the African populations along the southern axis, we observe three major clusters: a two-member cluster representing the two Pygmy sub-groups; a three-member cluster comprising Bantu-speaking peoples from Kenya, Yorubans from Nigeria, and Mandenkas from Senegal; and finally a singleton cluster for the San from Namibia. These results are consistent with a recent phylogenetic study [28] that found the San to be the most isolated of the African populations, followed by the two Pygmy populations, and finally the three Bantu-language populations. Along the eastern axis,

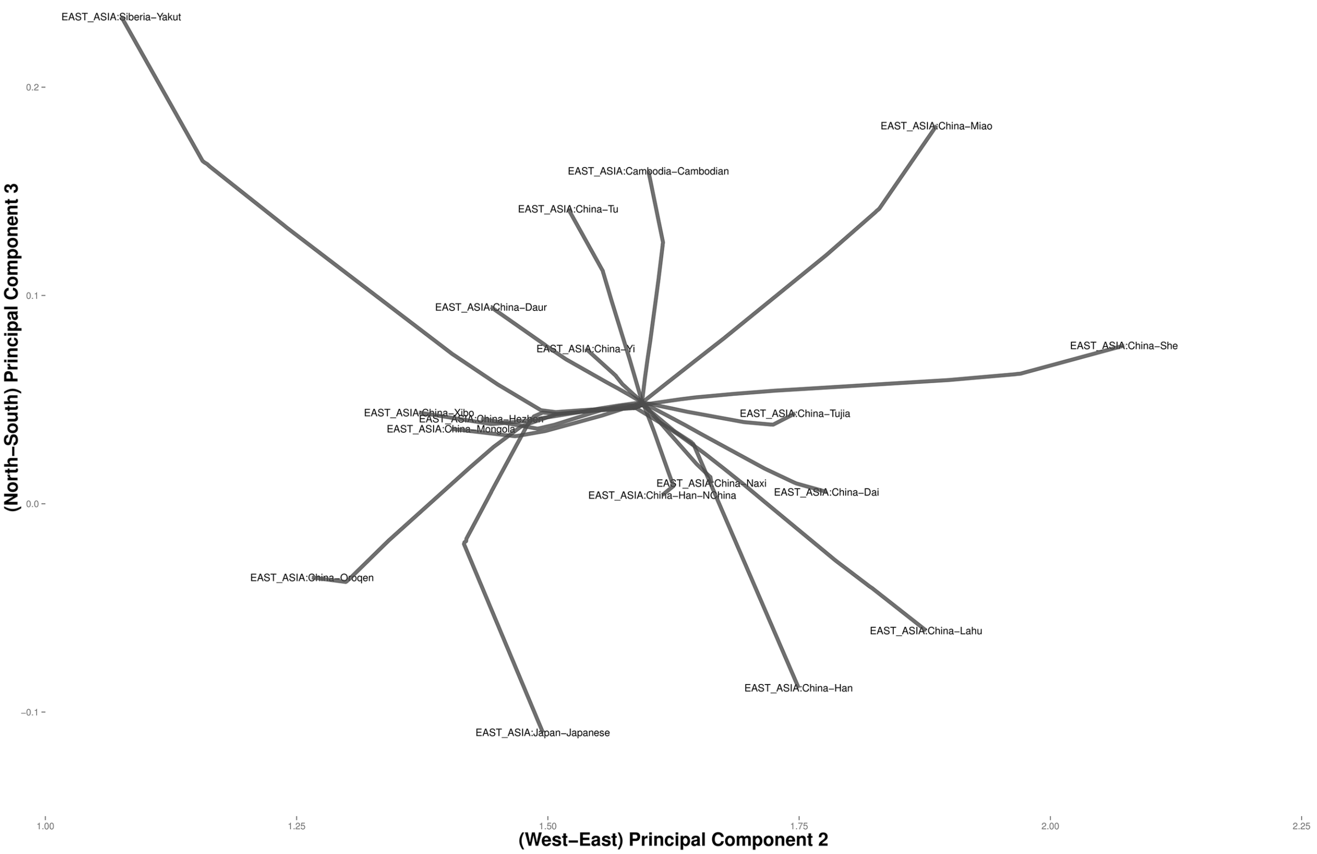

Fig 6. Magnified view of the convex clustering results for the HGDP data in East Asia.

doi:10.1371/journal.pcbi.1004228.g006

the two Papua New Guinea populations cluster together and do not join the remaining Asian populations.

Figs 6 and 7 focus on related populations along the eastern and northern axes of East Asia, respectively. Most of the Chinese populations along the eastern axis appear to coalesce simultaneously. Some of the other populations along the northern border of China coalesce earlier. The Hezhen and Oroqen peoples reside predominantly in the Heilongjiang province of northeast China [29, 30]. These two populations cluster early with the inner Mongolians and the Xibo population, who occupy northeast China and the northwest region of Xinjiang province. Three distinct clusters of Middle Easterners, Central Asians, and Europeans occur along the northern axis. All European populations except for the Russian populations are grouped into a single cluster. The two Russian populations instead merge with a second cluster that includes three populations from Israel. The Mozabites, who coalesce late with this cluster, exhibit high frequencies of North African haplotypes as previously noted in the literature [31, 32]. A third cluster within Central Asia unite Pakistani populations with Uygurs from China. Within this cluster, the Brahui, Balochi, and Makran populations of the Baluchistan province of northwestern Pakistan coalesce early with the Sindhi people of the Sindh province on the eastern border of Baluchistan. Later coalescing populations include the Hazara, Uygurs, and Kalash. The Hazaras of Pakistan and the Uygurs of China share common Mongolian and Turkic ancestry and some physical attributes [33, 34]. In contrast, hierarchical clustering suggests a more

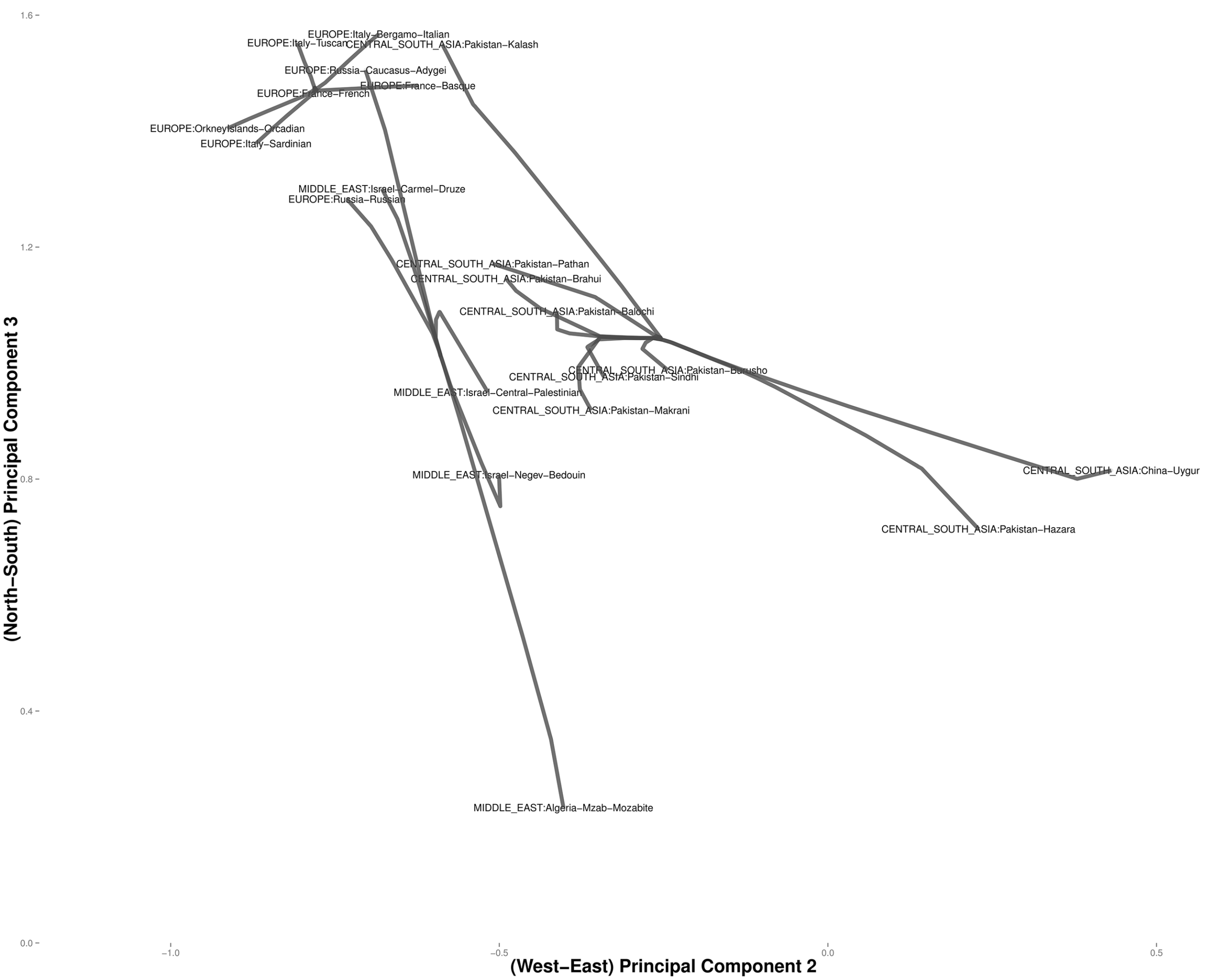

Fig 7. Magnified view of the convex clustering results for the HGDP data in Europe and Central Asia.

doi:10.1371/journal.pcbi.1004228.g007

distant relationship between these two ethnic groups. (Fig 8) A previous admixture analysis carried out on high-density SNP data via STRUCTURE [22] supports our observation that the Kalash people constitute a single distinct cluster, one of seven clusters separating all of the populations covered in the HGDP data [31].

Population structure of Europe. We next investigate whether convex clustering can glean further insights into the population structure of Europe. The POPRES resource archives high-density genotypes generated on the Illumina 550k microarray platform [35]. Version 2 of POPRES contains genotype and phenotype data on 4,077 subjects genotyped across 457,297 SNPs. For this analysis, we include only non-admixed Europeans who report all four grandparents of the same ethnicity. This leaves 1,896 subjects. SNP data presents advantages and disadvantages compared to microsatellite data. Dense marker panels may be more sensitive to subtle differences driven by population events such as migration, expansion, and bottlenecks [36].

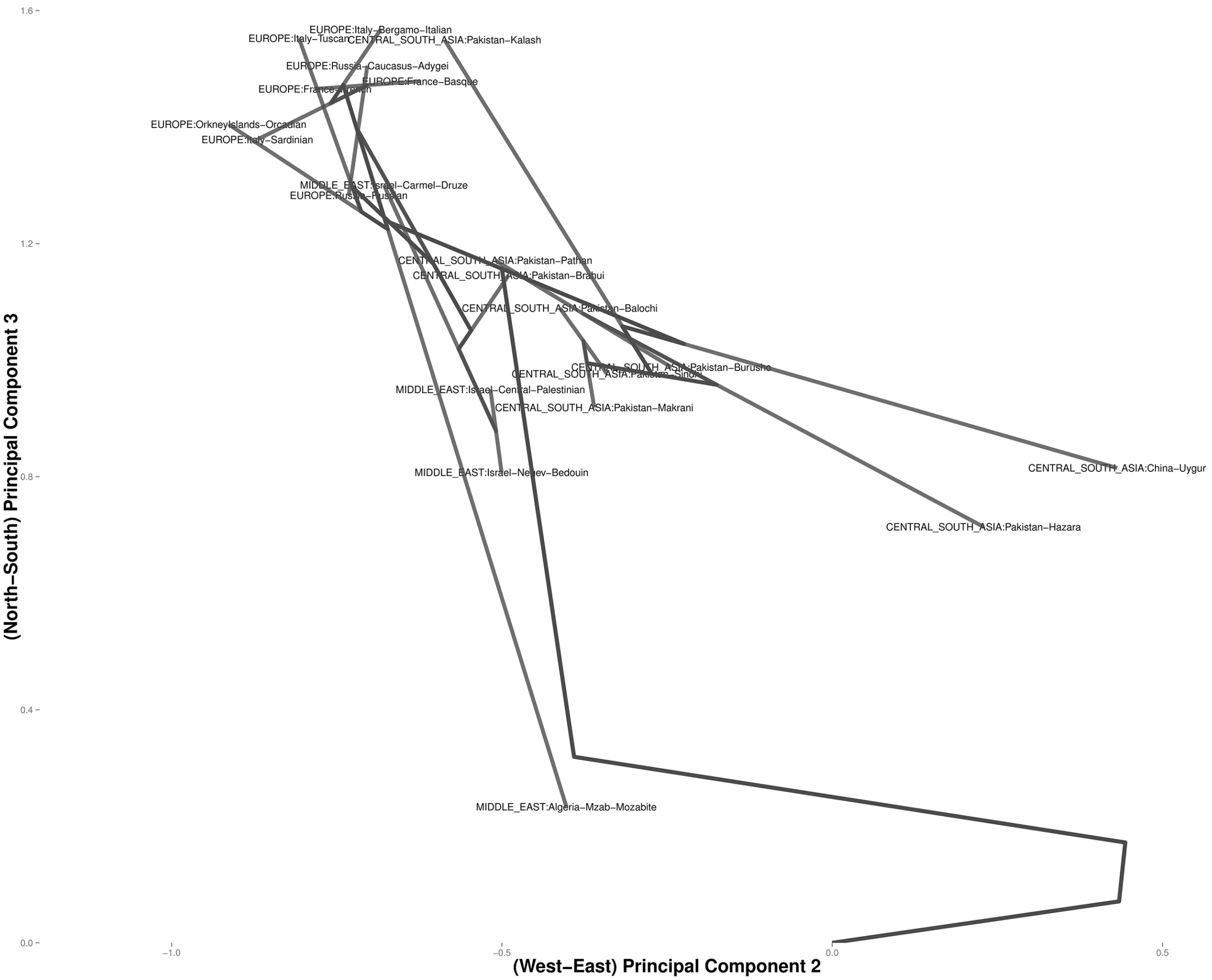

Fig 8. Magnified view of the hierarchical clustering results for the HGDP data in Europe and Central Asia.

doi:10.1371/journal.pcbi.1004228.g008

Challenges include the lower information content of biallelic markers and the correlations between markers caused by linkage disequilibrium (LD). After considerable experimentation, we found that the leading principal components offered more insight into population structure than the raw genotypes themselves. We employed EIGENSTRAT to extract the ten leading principal components from the genotype matrix. EIGENSTRAT prunes SNPs in LD with r^2 exceeding a user-specified threshold [24]. In our case the threshold 0.8 discards all but 276,823 nearly independent SNPs. Our choice of the composite distance defined in Eq (10) places equal weight ($\alpha = 0.5$) on genetic distances and GPS distances between the capital cities of participants. To ease visualization, our figures display a maximum of 20 subjects from each ethnicity, for a total of 370 subjects. The computed convex clustering path is projected onto the first two principal components of the POPRES data; these components capture geographic east-west and north-south axes, respectively.

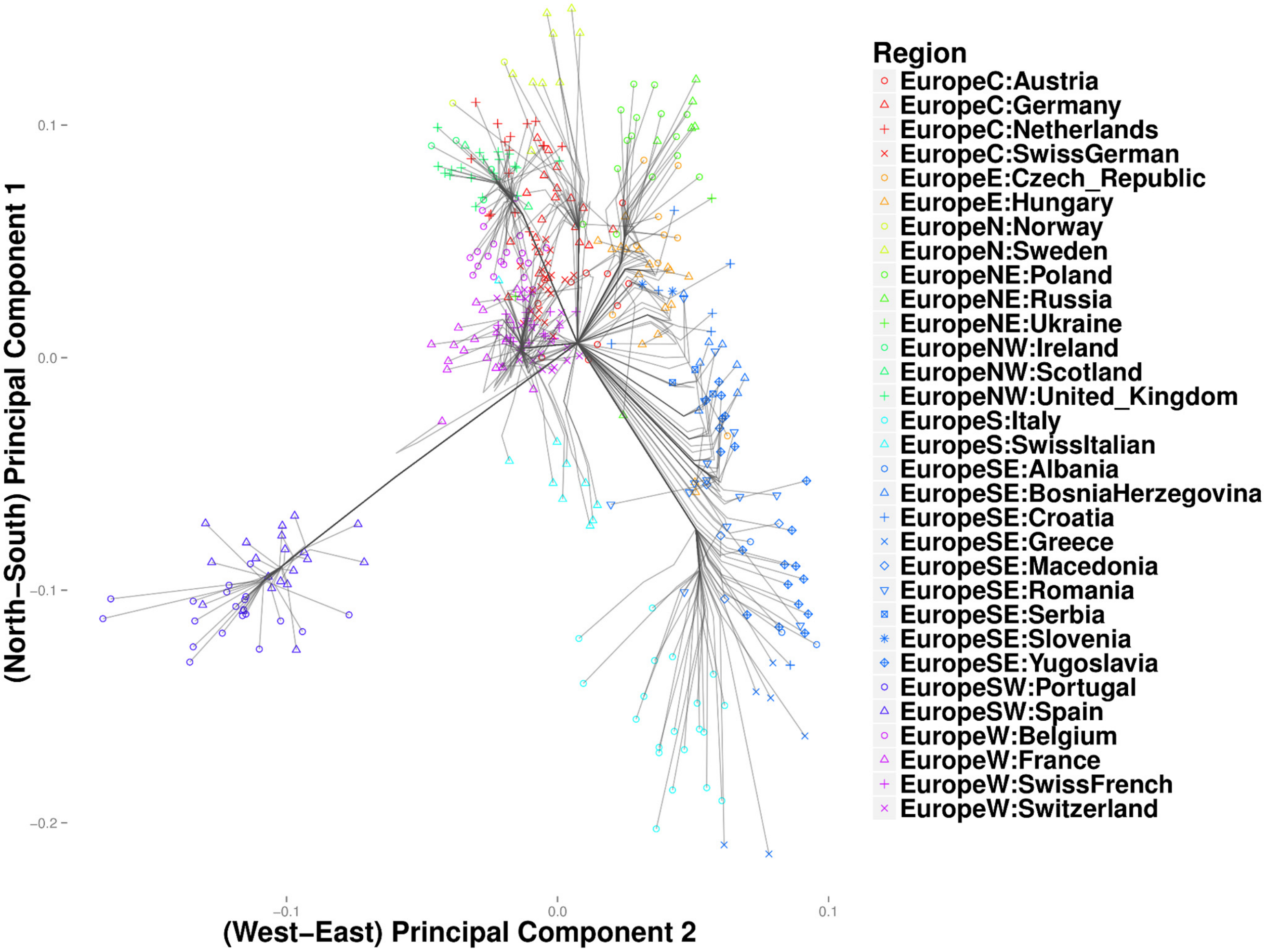

Fig 9. Convex clustering of the European populations from the POPRES data using $\phi = 0$ and $k = 40$.

doi:10.1371/journal.pcbi.1004228.g009

In the Iris and the HGDP datasets, the number of nearest neighbors k was more critical in resolving cluster evolution than the tuning constant ϕ. In the European POPRES data, where inter-class differences are more subtle, increasing ϕ can be critical in resolving details for k large. As in the previous examples, we gradually reduced k from a large value until major clusters along the North-West, North-East, and South-East geographic axes emerged. Fig 9 depicts a clustering path with $k = 40$ neighbors and $\phi = 0$. Increasing ϕ to 10 gives a similar clustering pattern, except that each of the major trunks coalesce before converging to the origin. Thus, Fig 10 shows several major clusters connected by five major trunks. Spain and Portugal constitute a major cluster in the southwest trunk. The southeast trunk includes Italy and southeast Europe; these populations eventually merge into a single cluster. The northeast trunk defines a cluster that includes Poland, Russia, Ukraine, the Czech Republic, Hungary, and Slovenia. Norway, Sweden, and Germany cluster along the northern trunk, and the British Isles merge with Belgium and the Netherlands to form the northwest trunk. A large cluster comprising France and the Swiss linguistic groups (French, German, and Italian) constitute the western trunk.

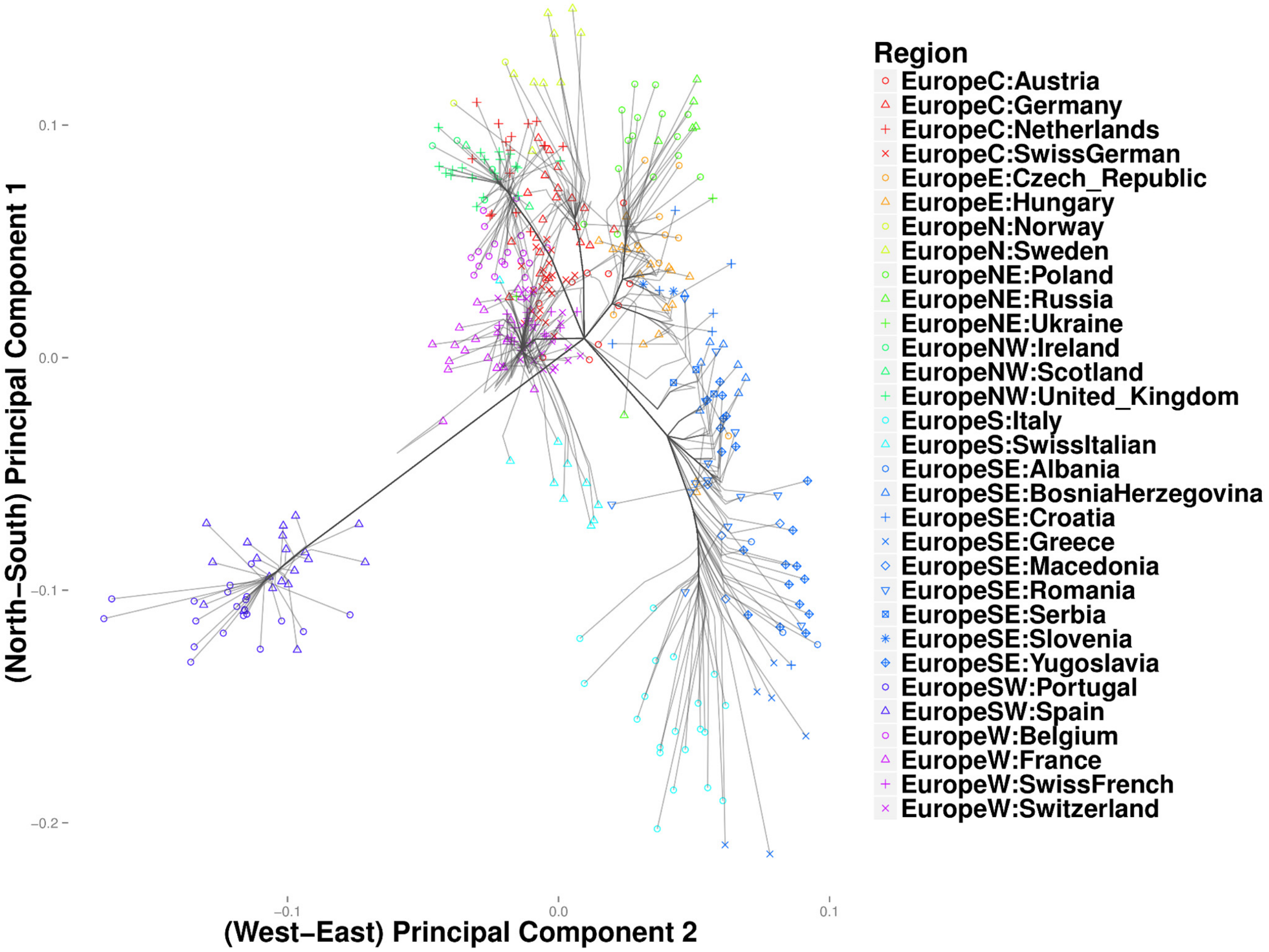

Fig 10. Convex clustering of the European populations from the POPRES data using $\phi = 10$ and $k = 40$.

doi:10.1371/journal.pcbi.1004228.g010

Hierarchical clustering for the most part recapitulates these major clusters, but the major clusters are less discernible. (Fig 11) Replotting the clustering path from convex clustering with ϕ = 1 and k = 3 shows Norway and Sweden breaking away from Germany and forming their own disjoint cluster (Fig 12). France breaks away from the Swiss groups to form its own disjoint cluster. Along the south trunk, Italy now separates from southeast Europe and eventually clusters with the Swiss-Italians.

Fig 13 depicts the clustering path of southeast Europe, where West Slavic languages predominate. Here Greece first coalesces with Macedonia, a Slavic population bordering Greece on the north. A cluster comprising Bosnia-Herzegovina and Serbia merges with Romania, before merging into the primary trunk of southeast Europe. Finally at the northern end of the trunk, a cluster formed by Croatia and Slovenia form its own cluster. The groups in the Bosnia-Herzegovina cluster and the Macedonian cluster are consistent with the recent break up of Yugoslavia. Poland and Russia cluster in the northern most branch of the northeast trunk (Fig 14). The Czech-Republic, Austria, and Hungary define a distinct cluster along the southern

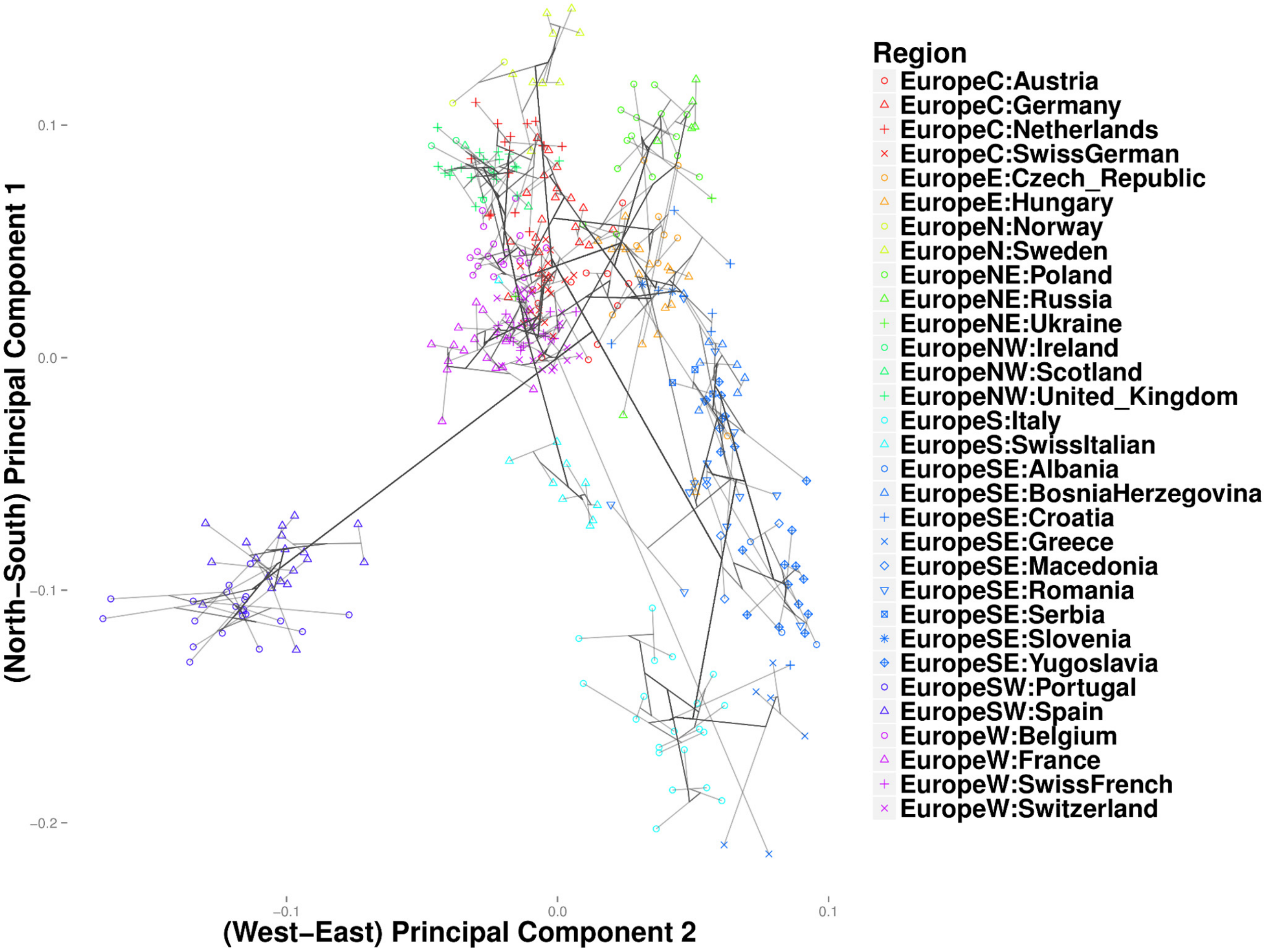

Fig 11. Hierarchical clustering of the European populations from the POPRES data.

doi:10.1371/journal.pcbi.1004228.g011

branch. Given that Austria conquered Hungary in 1699 and established rule over Bohemia (the predecessor to modern Czechs) as early as 1526, these results are not surprising.

In the POPRES data, convex clustering and hierarchical clustering occasionally disagree. For example, hierarchical clustering merges the Netherlands and Belgium with Britain before it merges Britain with Ireland and Scotland (Fig 15). In light of the geography and history of Britain, it is reasonable to expect Britain to first merge with Scotland and Ireland. Convex clustering produces yields precisely this expected effect (Fig 16). The British-Scotland-Ireland cluster then merges with the neighboring cluster of Belgium and the Netherlands. Owing to a few outliers, the greedy nature of hierarchical clustering appears to force a spurious coalescence, which cannot be repaired until later. Another discrepancy occurs in clustering the Swiss linguistic groups. Convex clustering first groups the Swiss-German, Swiss-French, and Swiss-Italian into a single Swiss cluster (Fig 17). Hierarchical clustering groups France with this cluster. At the next higher level, rather than cluster Italy with the Swiss, hierarchical clustering merges it with

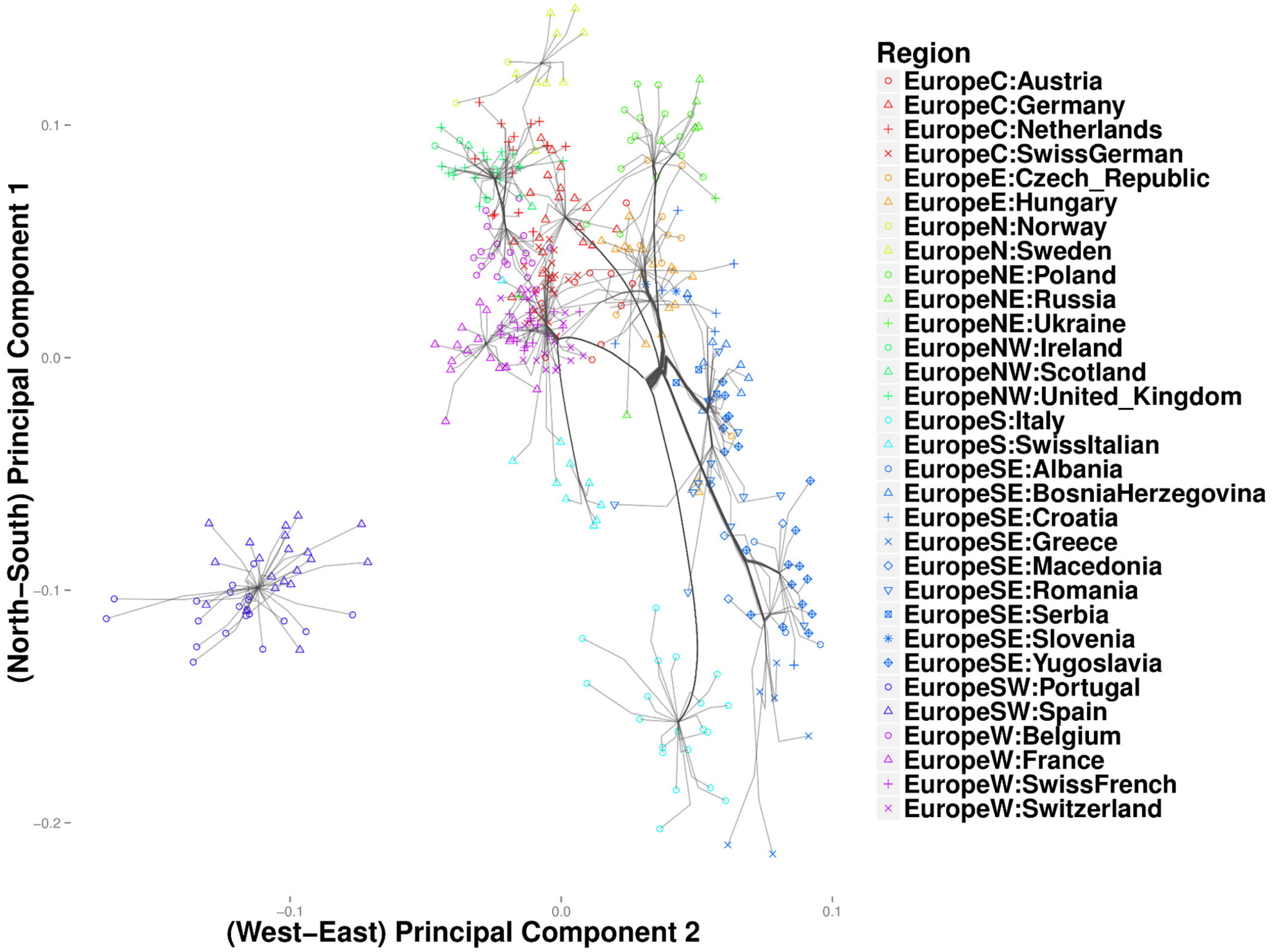

Fig 12. Convex clustering of the European populations from the POPRES data using $\phi = 1$ and $k = 3$.

doi:10.1371/journal.pcbi.1004228.g012

Greece and populations from the former Yugoslavia. Convex clustering, in contrast, merges Italy with the Swiss before joining both to the southeast European trunk. In this case, it is unclear which method is providing a more accurate solution; due to the large size of Italy, geographic proximity suggests a closer relationship between Southern Italians and Greece, with similar logic applied to Northern Italians and the Swiss. Further details on the geographic origins of POPRES Italian subjects would help resolve this discrepancy.

Inferring Cancer Subtypes

It is well accepted that cancers of a given tissue often fall into different subtypes. In breast cancer for instance, patients with tumors that are estrogen receptor (ER) and epidermal growth factor receptor (ErbB2) negative are less responsive to hormone based treatment than those possessing active receptors [37]. High-throughput platforms such as gene-expression microarrays and RNA-Seq have enabled researchers to classify cancer patients based on their molecular phenotypes. Hierarchical clustering by [38] established five gene-expression profiles across

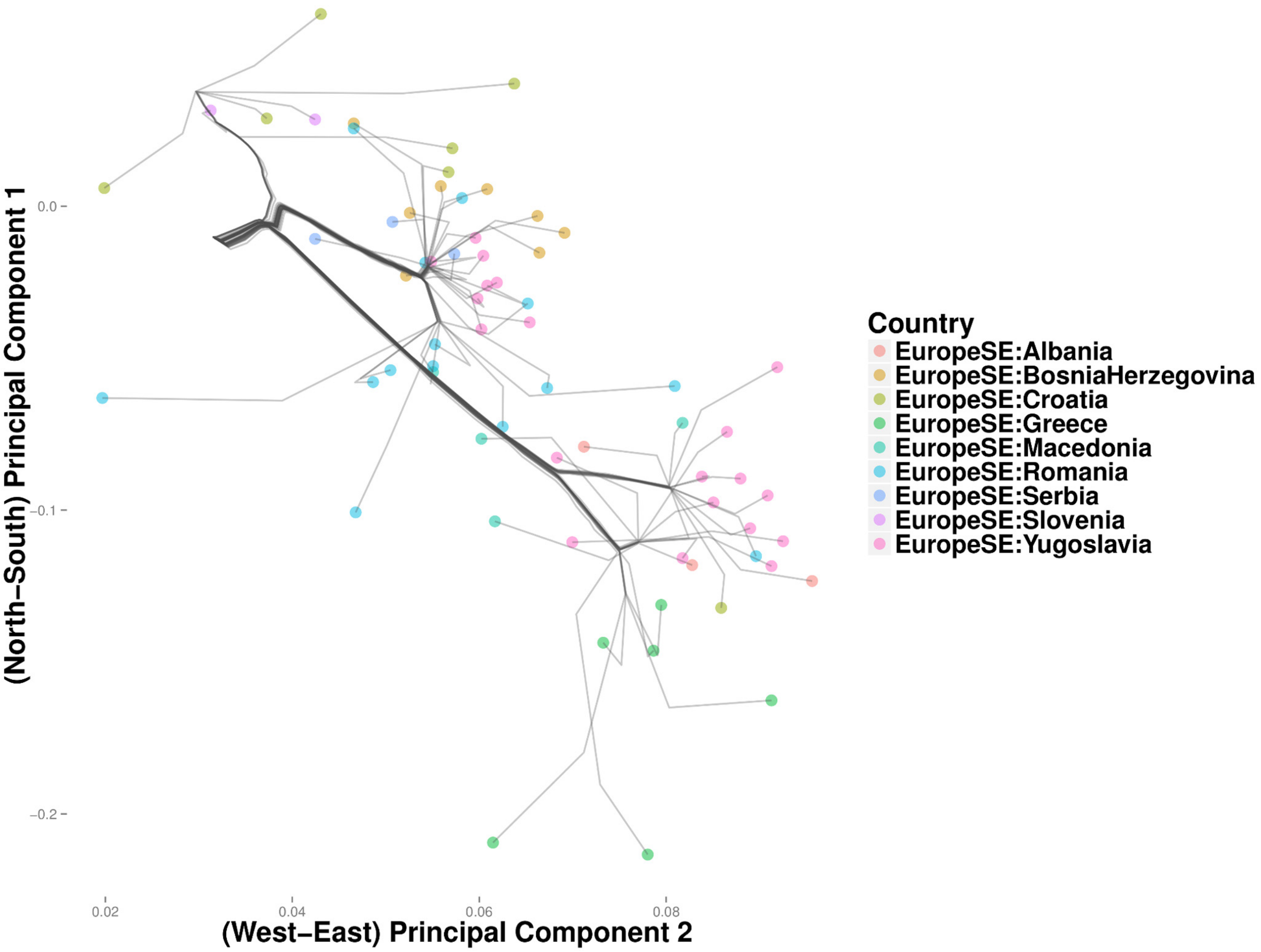

Fig 13. Magnified view of results from convex clustering of Southeast Europe.

doi:10.1371/journal.pcbi.1004228.g013

9216 genes in 84 breast-cancer patients. Among the 84 patients, only 16 also had a clinical assessment of hormone receptor status. Here, we attempt to determine whether convex and hierarchical clustering can infer clusters consistent with the clinical outcomes for these 16 patients. Under the tuning constants $\phi = .5$ and $k = 1$, convex clustering recovers two distinct clusters. Fig 18 projects the cluster centers along the cluster path on the first and third principal components of the original data. The left and right clusters correspond roughly to ER positive and ER negative tumors, respectively. Two ER negative tumors cluster with the ER positive tumors. Fig 19 depicts results from hierarchical clustering. Based on the order of fusion events, hierarchical clustering does not appear to group the tumors into distinct ER positive and negative groups. This could be an artifact of the hard binary choices imposed by hierarchical clustering. The two ER-B2 positive samples that clustered together in convex clustering appear in distant clusters under hierarchical clustering.

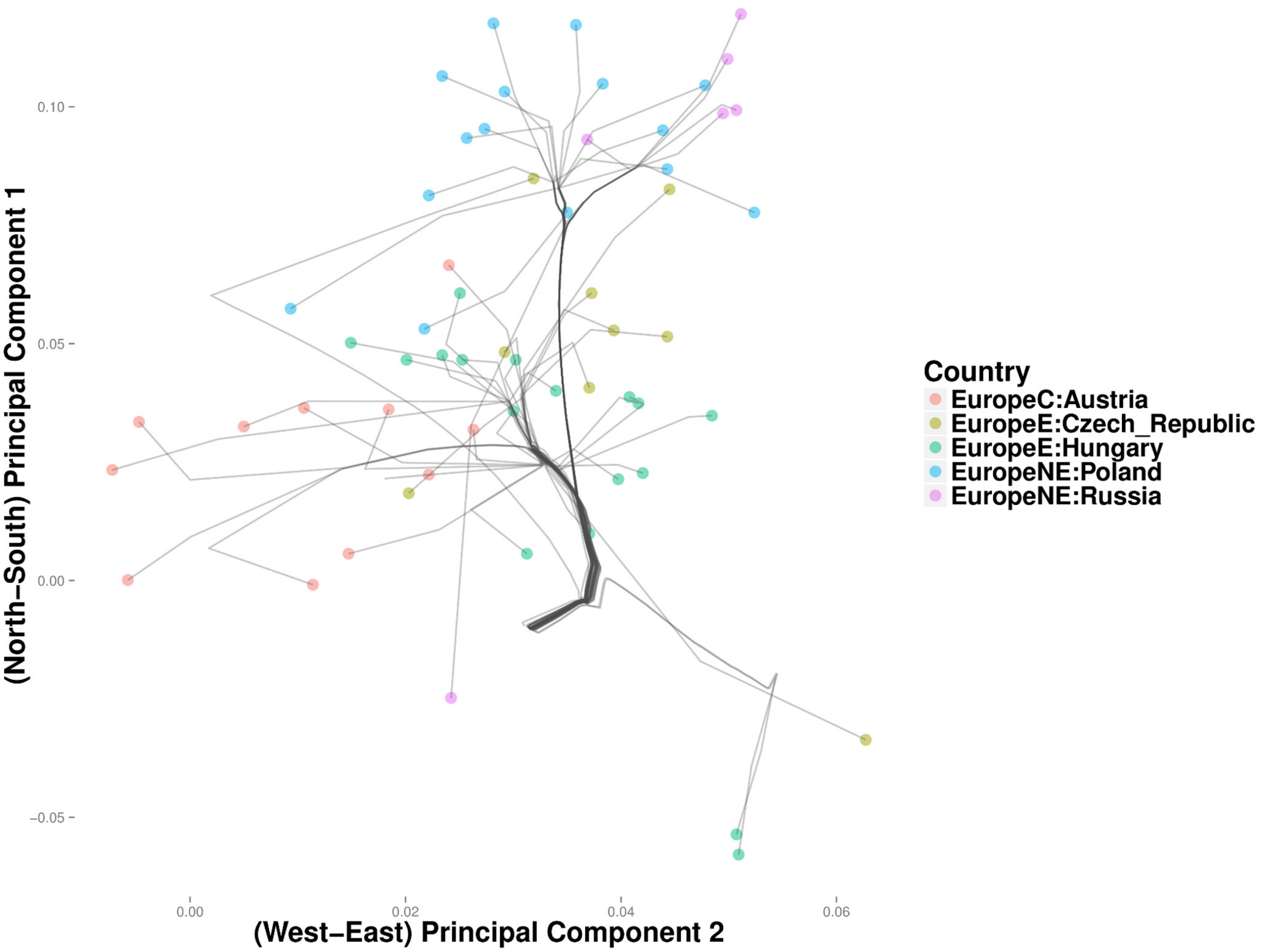

Fig 14. Magnified view of results from convex clustering of Northeast Europe.

doi:10.1371/journal.pcbi.1004228.g014

Run-Time Benchmarks

For a dataset with a large number of attributes, parallelization can substantially reduce run times. CONVEXCLUSTER includes code written in OpenCL, a language designed to run on many-core devices such as GPUs. For each of the three genetic analyses presented above, we recorded the total run-time along the entire regularization path using standard C++ code for the CPU and OpenCL code for the GPU. For the sake of comparison, we also recorded run-times for CLUSTERPATH [3], an R package that also implements convex clustering, on the same datasets and weighting schemes. Table 3 records the average run time to minimize the objective function averaged over all values of the regularization parameter. We chose this strategy because CLUSTERPATH does not allow users to pre-specify a grid of regularization values. The bottom line is that CONVEXCLUSTER required only 16%, 47%, and 75% of the time required by CLUSTERPATH to fit the HGDP, POPRES, and breast cancer datasets respectively. When a GPU is available, further improvements can potentially be realized. On an nVidia C2050 GPU, CONVEXCLUSTER

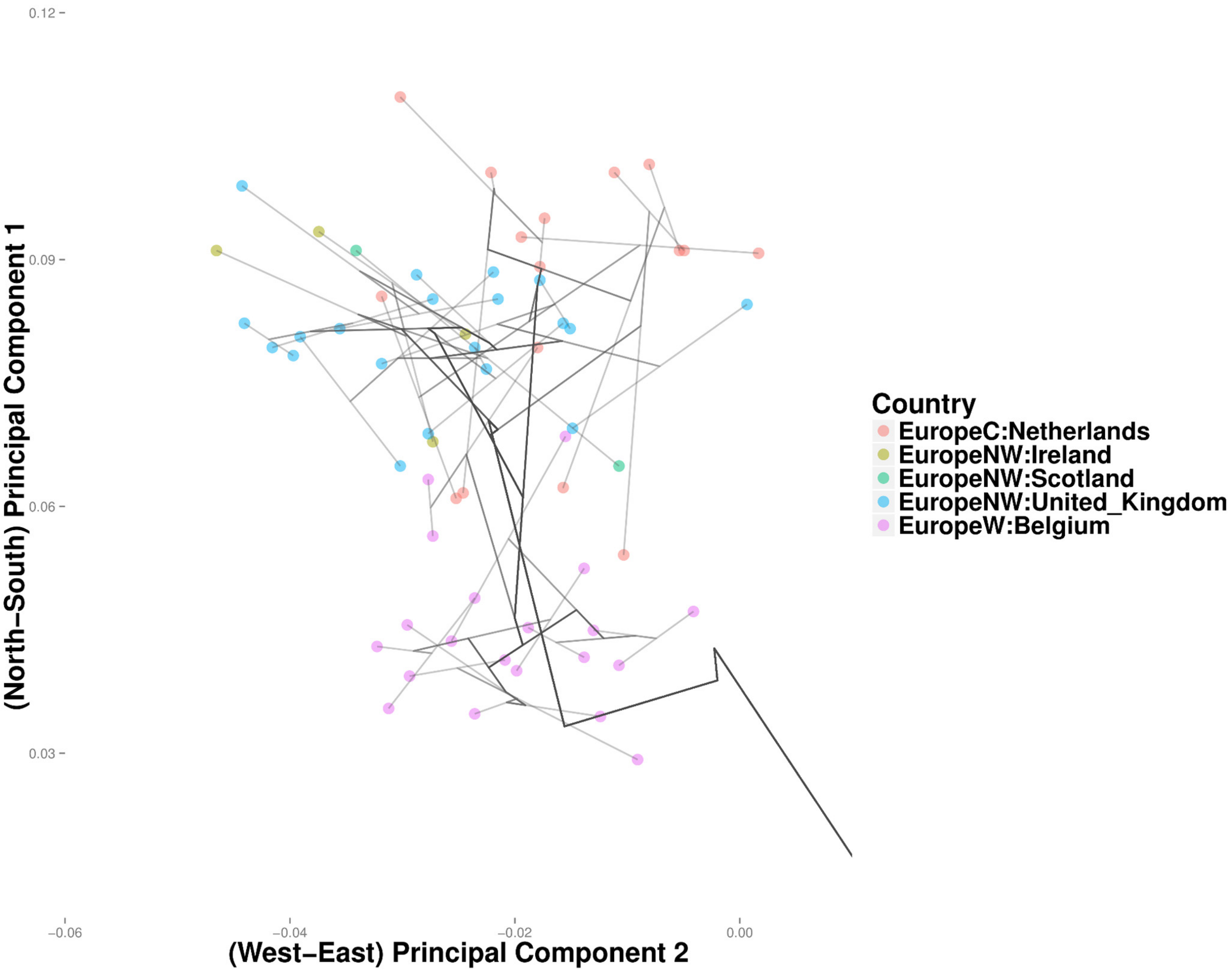

Fig 15. Hierarchical clustering projection showing genetic relationships among populations in and near the British Isles.

doi:10.1371/journal.pcbi.1004228.g015

enjoys speed improvements of 4.6 and 5.5 fold over the CPU version for the HGDP and breast cancer examples. In contrast, on the POPRES example, the GPU version is actually 3.5 fold slower than the CPU version. In its current form, CONVEXCLUSTER reads the updated matrix $\boldsymbol{U}$ from the GPUs at each point on the μ-regularization path before saving the data to disk. This large I/O overhead can overwhelm gains from parallelization for low-dimensional datasets such as the POPRES data. In general, GPU implementations of standard algorithms require a high degree of parallelization, limited data transfers between the master CPU and the slave GPUs, and maximal synchrony of the GPUs. Depending on the nature of the clustering data, CONVEXCLUSTER satisfies these requirements. It does not in the POPRES data, and computational efficiency suffers in the GPU version.

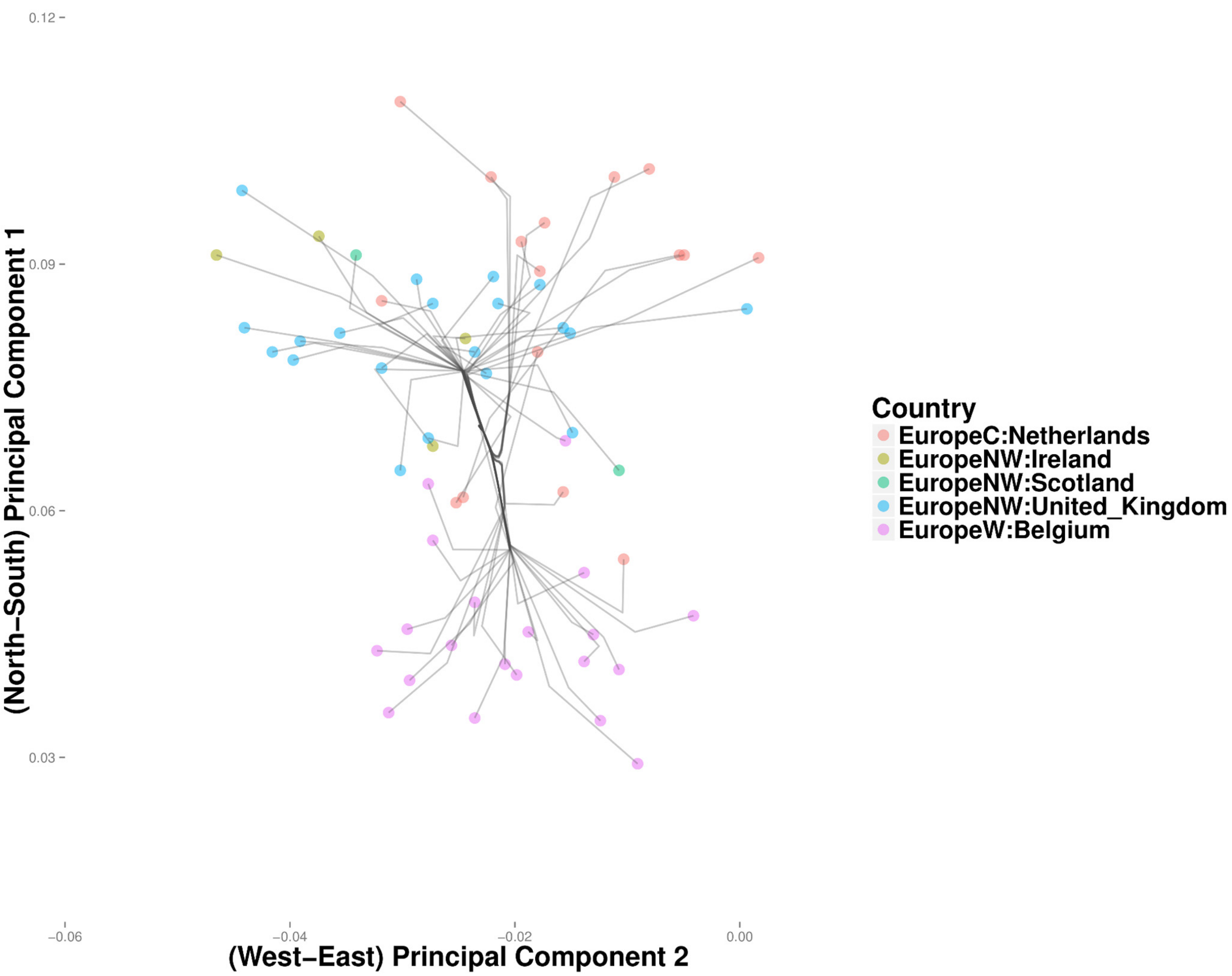

Fig 16. Convex clustering projection showing genetic relationships among populations in and near the British Isles.

doi:10.1371/journal.pcbi.1004228.g016

Discussion

The literature on cluster analysis is enormous. Each clustering method has advantages in either simplicity, speed, reliability, interpretability, or scalability. If the number of clusters is known in advance, then *k*-means clustering is usually preferred. In convex clustering one can often achieve a predetermined number of clusters by varying the number of nearest neighbors and following the solution path to its final destination. Alternatively, if the underlying graph is fully connected, then one can follow the solution path until *k* clusters appear. The downside of *k*-means clustering is that it offers no insight into cluster similarity. If the goal in clustering is to obtain a snapshot of the relationships among observed data points at different levels of granularity, the choices are limited, and most biologists opt for hierarchical clustering. Hierarchical clustering is notable for its speed and visual appeal. Balanced against these assets is its sensitivity to poor starting values and outliers. Convex clustering occupies an enviable middle ground

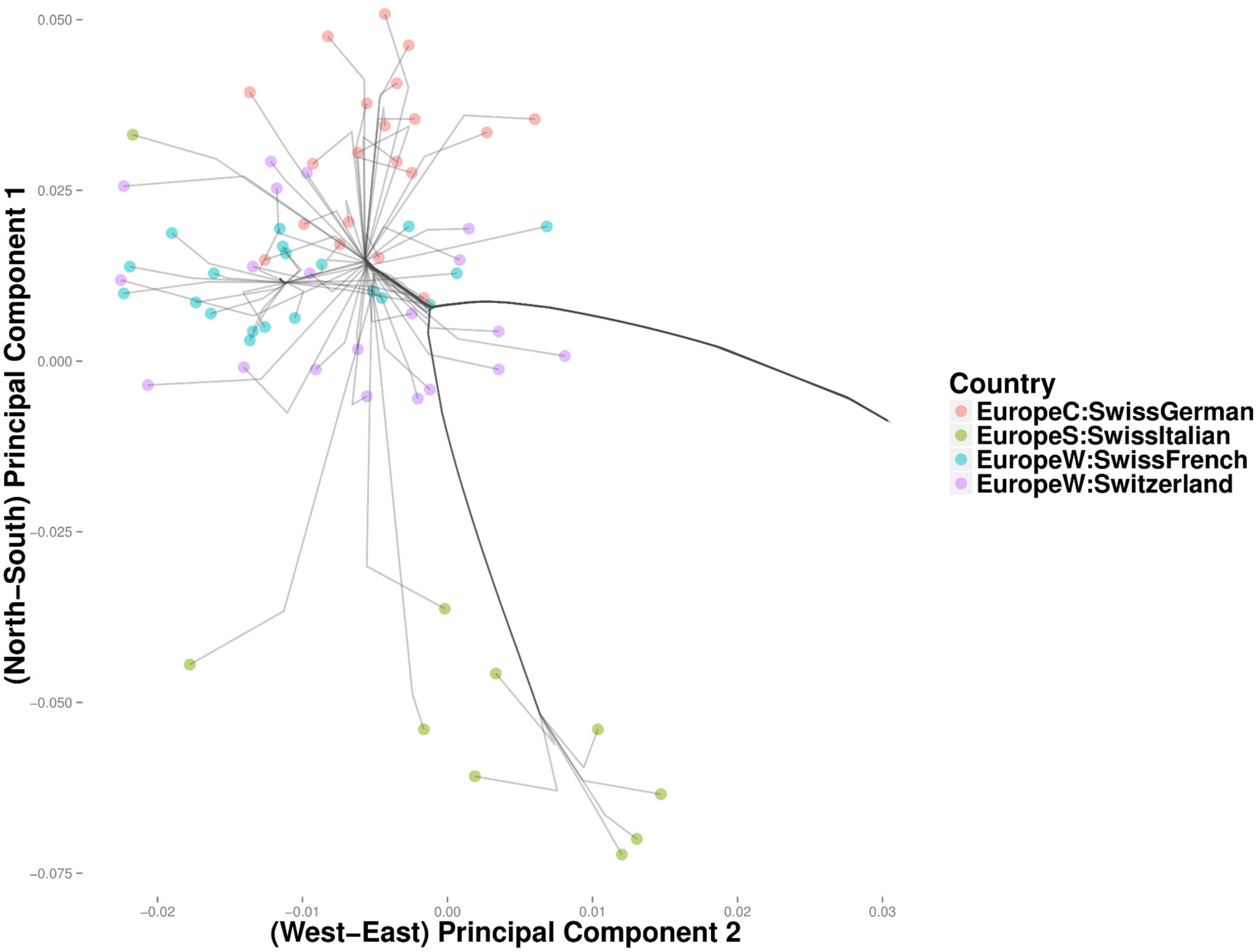

Fig 17. Magnified view of results from convex clustering of Swiss liguistic groups.

doi:10.1371/journal.pcbi.1004228.g017

between *k*-means clustering and hierarchical clustering. Our extensive exploration of the HGDP and POPRES datasets showcase the subtle solutions paths of convex clustering. These paths offer considerable insights into population history and correct some of the greedy mistakes of hierarchical clustering.

Nonetheless, hierarchical clustering can be the more practical choice when noise is low and a premium is put on computational speed. In the Iris data with no introduced noise, the two methods yield equivalent results. Total runtimes for the convex clustering analyses in this paper ranged from 5 minutes to 30 minutes. In contrast, even for the largest datasets analyzed here, hierarchical clustering required no more than 5 seconds to complete. Our perturbations of the Iris data demonstrate sensitivity to noise, so speed comes at a price.

Given the novelty of convex clustering [2], it is hardly surprising that only a few previous programs (CLUSTERPATH [3] and CVXCLUSTR [39]), implement it. Our program is unique in that we offer a fast implementation when GPU devices are available. These earlier programs

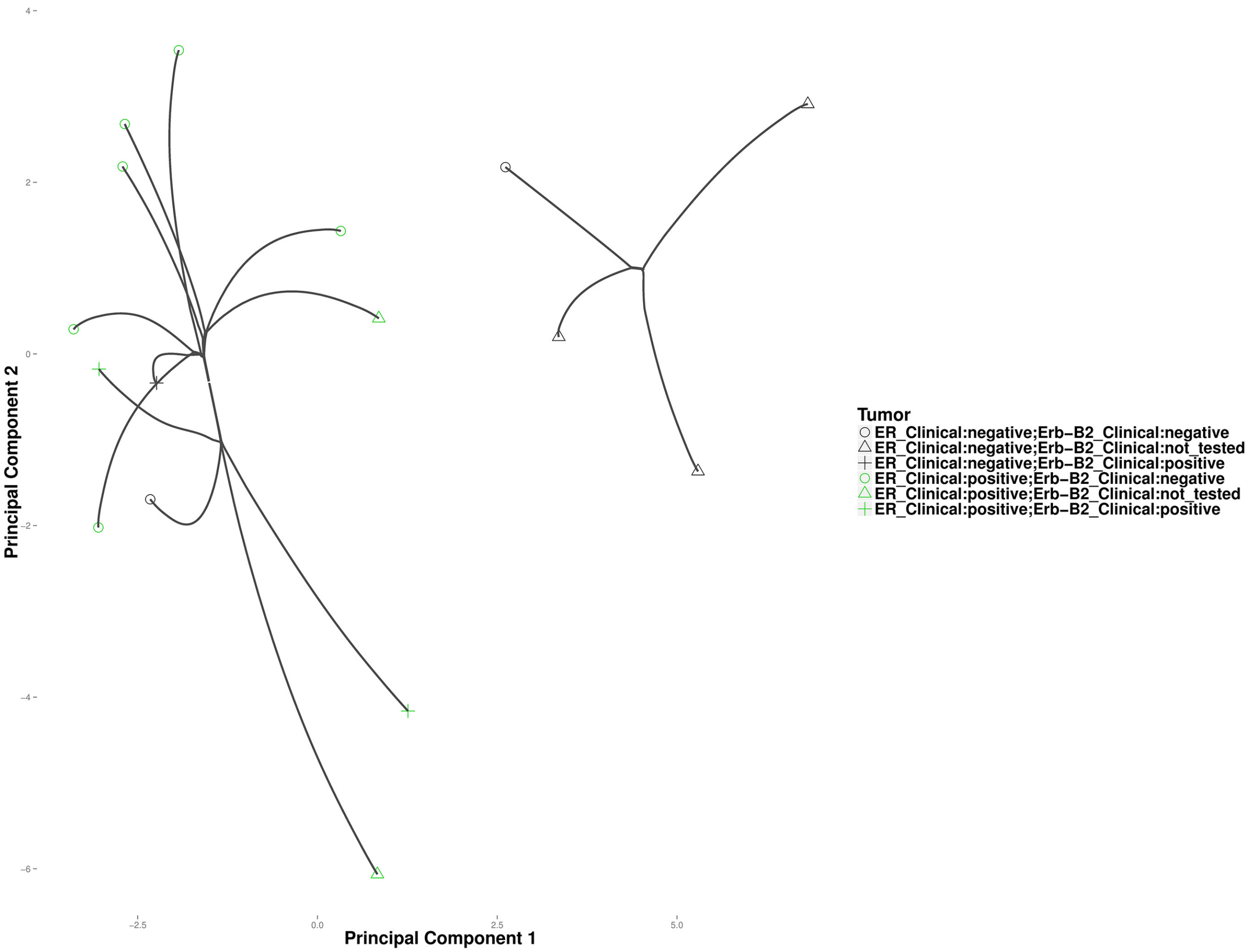

Fig 18. Convex clustering of the breast cancer samples. Points on the plot indicate data vectors projected onto the first and third principal components (PCs) of the sample. Lines trace the cluster centers as they traverse the regularization path.

doi:10.1371/journal.pcbi.1004228.g018

perform similarly to our program on modest problems such as the Iris data. Unfortunately, on large datasets such as the HGDP data, CLUSTERPATH depletes all available memory and terminates prematurely. Furthermore, CLUSTERPATH lacks two features that work to the advantage of convex clustering. First, it does not support disconnected graphs defined by sparse weights. In our breast cancer example, clustering with disconnected graphs reveals fine-grained details. Second, CLUSTERPATH does not allow for missing entries in the data matrix. The current paper documents CONVEXCLUSTER's ability to scale realistically to dimensions typical of modern genomic data. A combination of careful algorithmic development and exploitation of modern many-core chipsets lies behind CONVEXCLUSTER. The proximal distance algorithm propelling CONVEXCLUSTER separates parameters and enables massive parallelization. OpenCL made it relatively easy to implement parallel versions of our original serial code. Further speedups are possible. For instance, CONVEXCLUSTER spends an inordinate amount of execution time moving matrices over relatively slow I/O channels in preparation for plotting. One could easily project

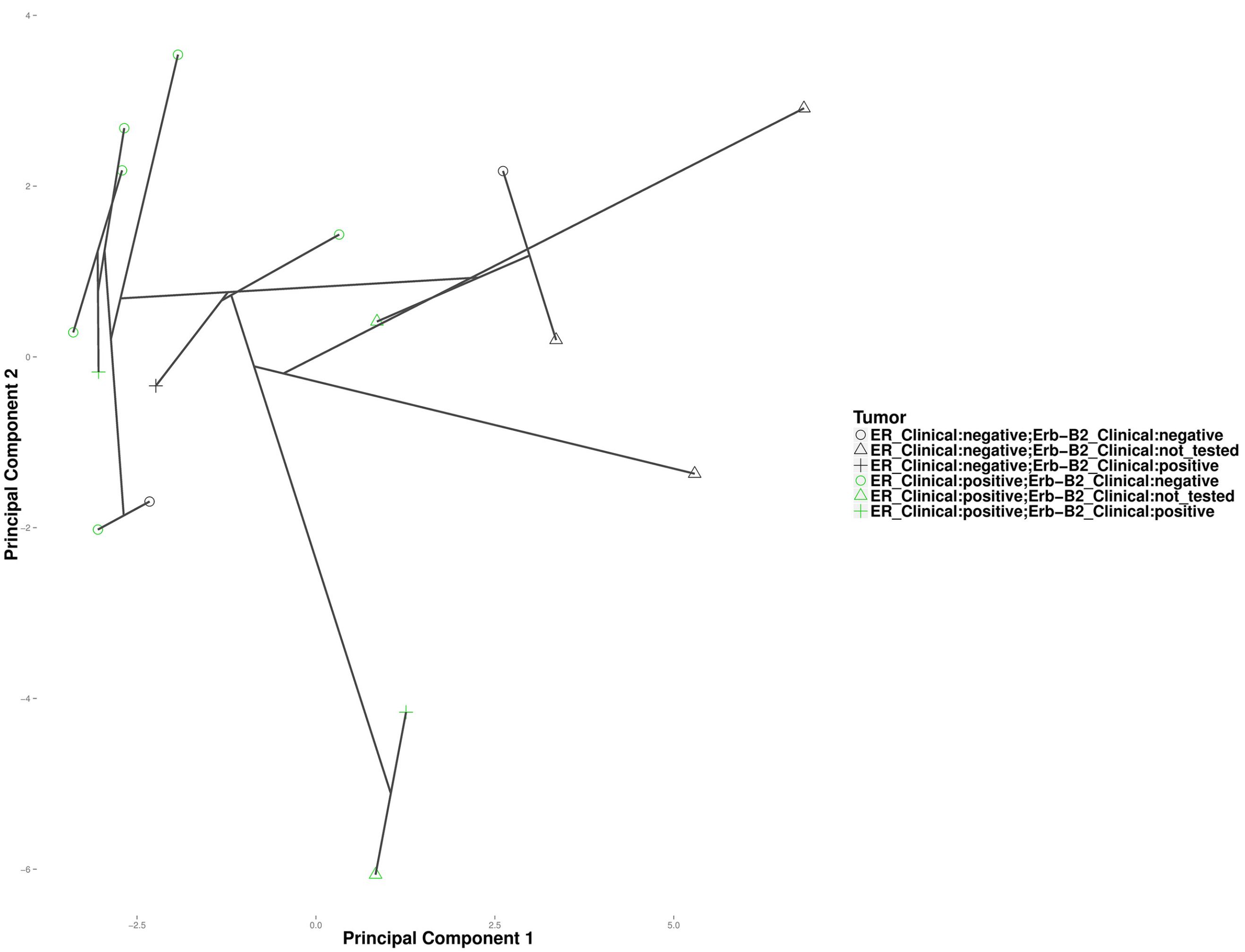

Fig 19. Average linkage hierarchical clustering of the breast cancer samples.

doi:10.1371/journal.pcbi.1004228.g019

Table 3. Average runtimes in seconds for different analyses.

Analysis	Datapoints	Variables	CLUSTERPATH	CONVEXCLUSTER	
				CPU	GPU
HGDP	52	4,682	8.67	1.46	.32
POPRES	370	10	2.53	1.21	4.29
Breast Cancer data	16	9,216	3.14	2.37	.43

doi:10.1371/journal.pcbi.1004228.t003

the data to principal components on each GPU itself prior to data transfer. More recent ATI or nVidia GPUs should improve the speedups on high-dimensional data mentioned here.

Convex clustering also shows promise as a building block for more sophisticated exploratory tools in computational biology. In a companion paper [40] introduce a convex formulation of the biclustering problem. In biclustering one seeks to cluster both observations and features simultaneously in a data matrix. Cancer subtype discovery can be formulated as a biclustering problem in which gene expression data is partitioned into a checkerboard-like pattern highlighting the associations between groups of patients and the groups of genes that distinguish them. To bicluster a data matrix, hierarchical clustering can be applied independently to the rows and columns of the matrix. Convex biclustering produces more stable biclusterings while retaining the interpretability of hierarchical biclustering. Convex biclustering requires repeatedly solving convex clustering subproblems.

The field of cluster analysis is crowded with so many competing methods that it would foolish to conclude that convex clustering is uniformly superior. Our goal of illustrating the versatility of convex clustering is more modest. The reflex reaction of most biologists is to employ hierarchical or *k*-means clustering. We suggest that biologists take a second look. Convex clustering's ability to reliably deliver an entire solution path is compelling. The insights discussed here will enhance the careful exploration of many big datasets. The present algorithm, and indeed the present formulation of convex clustering, are unlikely to be the last words on the subject. We encourage other computational biologists and statisticians to refine these promising tools. CONVEXCLUSTER can be freely downloaded from the UCLA Human Genetics web site at http://www.genetics.ucla.edu/software/ for analysis and comparison purposes.

Author Contributions

Conceived and designed the experiments: GKC ECC KL. Performed the experiments: GKC. Analyzed the data: GKC. Wrote the paper: GKC ECC KL. Prepared the POPRES data: JMOR.

References

1. Cormen TH, Stein C, Rivest RL, Leiserson CE. Introduction to Algorithms. 2nd ed. McGraw-Hill Higher Education; 2001.
2. Lindsten F, Ohlsson H, Ljung L. Clustering using sum-of-norms regularization: With application to particle filter output computation. In: Statistical Signal Processing Workshop (SSP), 2011 IEEE. IEEE; 2011. p. 201–204.
3. Hocking T, Vert JP, Bach F, Joulin A. Clusterpath: an Algorithm for Clustering using Convex Fusion Penalties. In: Getoor L, Scheffer T, editors. Proceedings of the 28th International Conference on Machine Learning (ICML-11). ICML '11. New York, NY, USA: ACM; 2011. p. 745–752.
4. Fisher RA. The use of multiple measurements in taxonomic problems. Annals of eugenics. 1936; 7 (2):179–188. doi: 10.1111/j.1469-1809.1936.tb02137.x
5. Lange K, Keys KL. The MM proximal distance algorithm. Proceedings 2014 International Congress of Mathematicians. 2014;(in press).
6. Borwein JM, Lewis AS. Convex Analysis and Nonlinear Optimization. vol. 3 of CMS Books in Mathematics. 2nd ed. Springer; 2006.
7. Clarke FH. Optimization and Nonsmooth Analysis. vol. 5 of Classics in Applied Mathematics. SIAM; 1990.
8. Demyanov VF, Fletcher R, Terlaky T, Di Pillo G, Schoen F. Nonlinear Optimization. Springer; 2010.
9. Borg I, Groenen PJ. Modern Multidimensional Scaling: Theory and Applications. Springer; 2005.
10. Heiser WJ. Convergent computation by iterative majorization: theory and applications in multidimensional data analysis. Recent Advances in Descriptive Multivariate Analysis. 1995;p. 157–189.
11. Hunter DR, Lange K. A tutorial on MM algorithms. American Statistician. 2004; 58:30–37. doi: 10.1198/0003130042836

12. Lange K, Hunter DR, Yang I. Optimization transfer using surrogate objective functions. Journal of Computational and Graphical Statistics. 2000; 9:1–20. doi: 10.2307/1390605

13. Wu TT, Lange K. The MM alternative to EM. Statistical Science. 2010; 25:492–505. doi: 10.1214/08-STS264

14. Deutsch F. Best Approximation in Inner Product Spaces. vol. 7 of CMS Books in Mathematics. Springer-Verlag; 2001.

15. Parikh N, Boyd S. Proximal algorithms. Foundations and Trends in Optimization. 2013; 1(3):123–231.

16. Lange K. Optimization. 2nd ed. Springer Texts in Statistics. Springer-Verlag; 2012.

17. Bache K, Lichman M. UCI Machine Learning Repository; 2013. Available from: http://archive.ics.uci.edu/ml.

18. Sokal RR, Michener CD. A statistical method for evaluating systematic relationships. University of Kansas Science Bulletin. 1958; 38:1409–1438.

19. Hopcroft J, Tarjan R. Algorithm 447: Efficient algorithms for graph manipulation. Communications of the ACM. 1973; 16(6):372–378. doi: 10.1145/362248.362272

20. Hubert L, Arabie P. Comparing partitions. Journal of Classification. 1985; 2(1):193–218. doi: 10.1007/BF01908075

21. Su YS, Gelman A, Hill J, Yajima M. Multiple Imputation with Diagnostics (mi) in R: Opening Windows into the Black Box. Journal of Statistical Software. 2011 12; 45(2):1–31. Available from: http://www.jstatsoft.org/v45/i02.

22. Pritchard JK, Stephens M, Donnelly P. Inference of population structure using multilocus genotype data. Genetics. 2000 Jun; 155(2):945–959. PMID: 10835412

23. Alexander DH, Novembre J, Lange K. Fast model-based estimation of ancestry in unrelated individuals. Genome research. 2009; 19(9):1655–1664. doi: 10.1101/gr.094052.109 PMID: 19648217

24. Price AL, Patterson NJ, Plenge RM, Weinblatt ME, Shadick NA, Reich D. Principal components analysis corrects for stratification in genome-wide association studies. Nat Genet. 2006 Aug; 38(8):904–909. doi: 10.1038/ng1847 PMID: 16862161

25. Rosenberg NA, Pritchard JK, Weber JL, Cann HM, Kidd KK, Zhivotovsky LA, et al. Genetic structure of human populations. Science. 2002 Dec; 298(5602):2381–2385. doi: 10.1126/science.1078311 PMID: 12493913

26. Kittles RA, Weiss KM. Race, ancestry, and genes: implications for defining disease risk. Annual Rev Genomics Hum Genet. 2003; 4:33–67. doi: 10.1146/annurev.genom.4.070802.110356

27. Wang S, Lewis CM Jr, Jakobsson M, Ramachandran S, Ray N, Bedoya G, et al. Genetic Variation and Population Structure in Native Americans. PLoS Genet. 2007 11; 3(11):e185. doi: 10.1371/journal.pgen.0030185 PMID: 18039031

28. Li JZ, Absher DM, Tang H, Southwick AM, Casto AM, Ramachandran S, et al. Worldwide human relationships inferred from genome-wide patterns of variation. Science. 2008 Feb; 319(5866):1100–1104. doi: 10.1126/science.1153717 PMID: 18292342

29. census bureau C. The Fourth Population Census of China in 1990; 1990.

30. census bureau C. Population Census of China in 2000; 2000.

31. Rosenberg NA, Mahajan S, Gonzalez-Quevedo C, Blum MG, Nino-Rosales L, Ninis V, et al. Low levels of genetic divergence across geographically and linguistically diverse populations from India. PLoS Genet. 2006 Dec; 2(12):e215. doi: 10.1371/journal.pgen.0020215 PMID: 17194221

32. Coudray C, Olivieri A, Achilli A, Pala M, Melhaoui M, Cherkaoui M, et al. The complex and diversified mitochondrial gene pool of Berber populations. Ann Hum Genet. 2009 Mar; 73(2):196–214. doi: 10.1111/j.1469-1809.2008.00493.x PMID: 19053990

33. Qamar R, Ayub Q, Mohyuddin A, Helgason A, Mazhar K, Mansoor A, et al. Y-chromosomal DNA variation in Pakistan. Am J Hum Genet. 2002 May; 70(5):1107–1124. doi: 10.1086/339929 PMID: 11898125

34. Ablimit A, Qin W, Shan W, Wu W, Ling F, Ling KH, et al. Genetic diversities of cytochrome B in Xinjiang Uyghur unveiled its origin and migration history. BMC Genet. 2013; 14:100. doi: 10.1186/1471-2156-14-100 PMID: 24103151

35. Nelson MR, Bryc K, King KS, Indap A, Boyko AR, Novembre J, et al. The Population Reference Sample, POPRES: a resource for population, disease, and pharmacological genetics research. Am J Hum Genet. 2008 Sep; 83(3):347–358. doi: 10.1016/j.ajhg.2008.08.005 PMID: 18760391

36. Gravel S, Henn BM, Gutenkunst RN, Indap AR, Marth GT, Clark AG, et al. Demographic history and rare allele sharing among human populations. Proceedings of the National Academy of Sciences. 2011; 108(29):11983–11988. Available from: http://www.pnas.org/content/108/29/11983.abstract. doi: 10.1073/pnas.1019276108

37. Rochefort H, Glondu M, Sahla ME, Platet N, Garcia M. How to target estrogen receptor-negative breast cancer? Endocr Relat Cancer. 2003 Jun; 10(2):261–266. doi: 10.1677/erc.0.0100261 PMID: 12790787

38. Perou CM, S?rlie T, Eisen MB, van de Rijn M, Jeffrey SS, Rees CA, et al. Molecular portraits of human breast tumours. Nature. 2000 Aug; 406(6797):747–752. doi: 10.1038/35021093 PMID: 10963602

39. Chi EC, Lange K. Splitting methods for convex clustering. Journal of Computational and Graphical Statistics. 2013.

40. Chi EC, Allen GI, Baraniuk RG. Convex Biclustering; 2014. arXiv:1408.0856 [stat.ME]. Available from: http://arxiv.org/abs/1408.0856.