Supplemental Methods

Additional Detail on Genotyping

The standard cluster file and output parameters were employed for export to a plain text final report file (<https://www.cottongen.org/data/community_projects/tamu63k#T1>).

Summary statistics were calculated for the number of markers passing filtering during each step using the “--het” and “--freq” commands in plink. Percent heterozygosity for each individual in each dataset was also calculated by dividing the number of observed heterozygous calls by the total number of calls.

Population Structure Analysis

The PCA implementation used from plink 1.9 was a direct port of EIGENSTRAT (Price et al. 2006), with normal reference/alternate allele coding and a built-in method to handle missing data. Double-centered PCA was performed using an R script from Gauch et al. (2019), with missing SNPs imputed with the median count of the minor allele. Eigenvalues were used to calculate percent variance explained by the first two dimension of PCA, calculated as the eigenvalue for the eigenvector divided by the sum of the eigenvalue for the first 40 eigenvectors. To reduce the effect of sign changing on the visual interpretation of PCA biplots, the PC1 vector was flipped by multiplying by -1 when necessary (Gauch et al. 2019).

Prior group assignment for DAPC was based on the original breeding group assignment. The plink bed format file was converted to a genind object using the “genomic\_converter” function in the R package radiator (Gosselin et al. 2020). A plink raw file, generated with the “--recode A” flag, was read in together with the map file with the “read.plink” command as a genlight object. DAPC was performed with the “dapc” command in interactive mode. To avoid model overfitting, the “optim.a.score” function was used to select the number of principal components. A DAPC biplot was generated using the original group numbers. The “compoplot.dapc” function was used to calculate and graph the assignment of individuals to each of the eight breeding groups.

For MEGA X, the concatenated DNA fasta file was generated by exporting from ped format with PGDSpider (Lischer and Excoffier 2012) and reading into MEGA X. The best DNA model was chosen using the minimum Bayesian information criterion “Find Best DNA/Protein Model” without invariable sites. A test of phylogeny was performed with the optimal model, the general time reversible model, and the bootstrap method with 1000 replicates.

The diagram for fastSTRUCTURE results was made by converting to a matrix object in R and plotting using the “compoplot” function in adegenet. The number of DAPC-derived groups was chosen using the minimum value of the Bayesian information criterion.

Core Collection Analysis

The number of homozygous SNP calls observed for each SNP allele was validated with a custom R script. The random sampling method was performed by first converting each row (SNP) in the SNP allele matrix (columns are individuals) into two rows, where each cell was either 1 for homozygous presence of that allele (one row for the major allele, one for the minor allele) or zero for heterozygous, no call, or homozygous for the other allele. Each row was then reduced to a single value, the probability of selecting an individual that was homozygous for that given SNP allele. The fiftieth percentile of the poisson binomial distribution was calculated using the “qpoisbinom” function in the R package poisbinom (Olivella and Shiraito 2017). For the breeding group number-informed method, MAF was calculated within each group, and the probability of discovering each allele was calculated as the product of (1 - MAFGroup1) \* (1 - MAFGroup2) \* (1 - MAFGroup3) … , with the group order [1,2,3,4,5,6,7,8] taken and repeated cyclically for the number of individuals sampled in the cth iteration.

For the minimum number of SNPs needed to identify the PD individuals, a matrix was generated with individuals as rows and SNPs as columns, with 1 for homozygous minor alleles and 0 otherwise. The column with the highest MAF was chosen as the first SNP, and the column with the highest Euclidean distance was chosen second (using the “dist” function in R). The SNP column the most different (on average) from the existing, already chosen SNPs were then added successively until each individual (row) could be uniquely identified.

Description of Supplemental Files

* TableS1-S15: Excel spreadsheets of supplemental tables.
* Supplemental Figures 1-7

Anchoring Marker Probe Sequences to Reference Genome

Complete marker flanking sequences were downloaded from Hulse-Kemp et al. (2015). The strand orientation was flipped to match the strand indicated in the project file Illumina Genome Studio. The 50 nucleotide sequence upstream of each probe sequence was extracted and saved into a fasta file, with each sequence labeled as the corresponding project marker name. The UTX\_v2.1 reference genome assembly for *Gossypium hirsutum*(Chen et al. 2020) was downloaded from Phytozome (Goodstein et al. 2012). A local BLAST database was built with the “makeblastdb” command. The probe sequences were queried against the database with the “blastn” command. The strict set of matching BLAST hits were filtered to only those with a minimum match length of 45 or longer. A more lenient set was generated to include lower e-value matches with another run of “blastn.”

A custom python script was used to combine information from the F2 intraspecific genetic map presented in Hulse-Kemp et al. (2015), inter-marker correlations, and BLAST hits. First, reciprocal best matches were identified based on inter-marker correlation, such that pair of highly correlated markers were identified (R2 > 0.8). The markers were anchored to the reference genome if the highest e-value BLAST hits for both markers were within 5 Mb on the same chromosome. The markers were not anchored if the chromosome assignment disagreed with the linkage group assignment from the F2 map. A random subset of 20% of the already anchored markers were chosen to extend the number of anchored markers to those with high inter-marker correlation with an already anchored marker, further choosing the most likely BLAST hit between high quality choices. Next, the remaining markers with F2 map positions were allocated to the corresponding pseudomolecule and inserted only if there was at least one nearby marker already inserted that was correlated with that marker.

This left a few types of markers: 1) those with a disagreement between the lowest e-value BLAST hit and the chromosome assigned from the F2 map, 2) markers absent on the genetic map with competing best insertion positions based on inter-marker correlation and lowest e-value BLAST hit, and 3) markers that either lacked a high quality BLAST hit or were not highly correlated with a nearby marker. To identify the best fitting insertion point for each marker, a random marker was chosen repeatedly until all markers had been addressed. For each marker, a goodness of fit score was assigned to each BLAST hit, providing a better score to insertion points with anchored markers with high inter-marker correlations with the selected marker. The score was calculated as the sum product of pairwise R2 and 1/log10(distance between BLAST hit and anchored marker + 10). At first, only those markers with the lowest e-value BLAST hit and LD-based score were inserted until no more markers could be anchored. Accordingly, tie-breaking was enabled, which showed a preference to the LD score over the BLAST hit e-value. Once tie-breaking yielded no further anchored markers for markers that either had no good BLAST hits or had no correlation with already anchored markers, the low-quality BLAST hits were evaluated instead.

This process was repeated 1000 times for various thresholds of inter-marker correlations, chosen from a uniform distribution ranging from R2=0.2-0.79. The results from bootstrapping were filtered to include markers that were successfully anchored to any chromosome in at least 80% of trials and mapped at least 20% more to the most frequent choice than the second most frequent choice. Bootstrap output is given in *S1 Table 16: Probe Alignment Bootstrap Results*).

Marker Density

To explore changes in the distribution of SNP marker loci across potential MAF values, between 0.025 and .500, the “density.compare” function in the R package sm was used (Bowman and Azzalini 2018). The nonparametric test for density equality, using the “model=equal” flag, was also performed, using the optimal density parameter, *h*. Next, the “sm.density.compare” function was used to evaluate changes in SNP marker density across chromosomes for the mapped markers in datasets one and two. The same nonparametric test for density equality was used (\* indicates *p* < 0.05).

Fit Against Pedigree Data

The pairwise identity by state (IBS) genetic distance matrix was generated in plink 1.9 with the “--dist 1-ibs” command for datasets one and two. Expanded pedigrees were used to calculate the generalized numerator relationship matrix, a value proportional to the expected percentage of identity by descent (IBD) alleles between individuals, with the NumericwareN software (Kim et al. 2016). Goodness of fit between IBS measurements of the two datasets was estimated by plotting the two against each other, and regression statistics calculated using “lm” with the formula “plink\_IBS\_dist\_dataset2 ~ plink\_IBS\_dist\_dataset1.” To test for goodness of fit to each of the pairwise genetic matrices, regression analysis was performed on the observed SNP-based IBS genetic distance for datasets one and two as explained by expected IBD estimate for each pair of genotypes. Regression statistics were calculated using the “lm” function in R, with the formula “plink\_IBS\_dist\_datasetn ~ NumericwareN\_IBD.”

**Code used for data analysis.**

* combiner.py/make\_map.py: Python3 script used to choose BLAST alignment for probe sequences
* sweeps.zip: file containing all inputs and run information for bayenv2
* analyzeIBS.R: R script used to compare IBS in Dataset One and Two and plot correlation plot
* analyzePCAs.R: R script used to make the PCA plots
* collectorsCurve.R: R script used to identify core SNPs, estimate allelic richness growth rate, and plot core collections from GenoCore
* corrVsNumnN.R + plotNumN-Corr: R script used to plot the correlation between NumericwareN coancestry and the genomic relationship matrix
* countUniqueAlleles.R: R script used to count the number of unique alleles in each breeding group
* doDAPC.R: R script to execute the DAPC command from adegenenet
* findDups.R: R script R script used to identify genotypes with > 97% similarity
* makeSankey.R: R script used to make the Sankey diagram
* plotBayEnv.R: R script used to plot the selective sweep information, in a Manhattan plot-type format
* plotCompoPlot.R: R script used to plot the composition plot for fastSTRUCTURE output
* plotDendrogram.R: R script used to plot the global dendrogram vs worldwide improved upland cotton
* plotDensity.R: R script used to plot allele density to find chromosomes with different SNP density between datasets
* plotDensityBarplot.R: R script used to plot total number of SNPs/Mb on each chromosome for each dataset
* plotFreqs.R: R script to plot the frequency of certain SNP alleles found through the sweeps
* plotNWK.R: R script to plot the phylogenetic tree from BLASTX
* plotVarWts.R: R script used to plot the variant weights by chromosome for PCA from Dataset One and Two.
* snpIntroductions.R: R script used to find the first group a SNP allele was introduced.
* plotDistTree.R: R script for plotting the NJT for the group-wise average IBS distance.
* collectorsCurve.R: R script for calculating the collector’s curve

REFERENCES

Bowman, A. W., and A. Azzalini, 2018 R package 'sm': nonparametric smoothing methods (version 2.2-5.6).

Chen, Z. J., A. Sreedasyam, A. Ando, Q. Song, L. M. De Santiago *et al.*, 2020 Genomic diversifications of five *Gossypium* allopolyploid species and their impact on cotton improvement. *Nature Genetics* **52** (5):525-533.

Gauch, H. G. J., S. Qian, H. P. Piepho, L. Zhou, and R. Chen, 2019 Consequences of PCA graphs, SNP codings, and PCA variants for elucidating population structure. *PloS One* **14** (6):e0218306.

Goodstein, D. M., S. Shu, R. Howson, R. Neupane, R. D. Hayes *et al.*, 2012 Phytozome: a comparative platform for green plant genomics. *Nucleic Acids Research* **40** (Database issue):D1178-1186.

Gosselin, T., M. Lamothe, F. Devloo-Delva, and P. Grewe, 2020 radiator: RADseq Data Exploration, Manipulation and Visualization using R.

Hulse-Kemp, A. M., J. Lemm, J. Plieske, H. Ashrafi, R. Buyyarapu *et al.*, 2015 Development of a 63K SNP Array for Cotton and High-Density Mapping of Intraspecific and Interspecific Populations of *Gossypium* spp. *G3 (Bethesda)* **5** (6):1187-1209.

Kim, B., W. D. Beavis, and J. Leon, 2016 Numericware N: Numerator Relationship Matrix Calculator. *Journal of Heredity* **107** (7):686-690.

Lischer, H. E., and L. Excoffier, 2012 PGDSpider: an automated data conversion tool for connecting population genetics and genomics programs. *Bioinformatics* **28** (2):298-299.

Olivella, S., and Y. Shiraito, 2017 R package 'poisbinom': A Faster Implementation of the Poisson-Binomial Distribution (version 1.0.1).

Price, A. L., N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick *et al.*, 2006 Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics* **38** (8):904-909.