Supplementary material for genome-wide genetic heterogeneity discovery with categorical covariates.

Felipe Llinares-López*, Laetitia Papaxanthos*, Dean Bodenham, Damian Roqueiro, COPDGene Investigators, Karsten Borgwardt

Contents

*Equally contributing authors.

S1 Supplementary Methods

S1.1 Implementation details

Algorithms 1 and 2 in the main manuscript present `FastCMH` using high-level pseudocode, which we reproduce here again for convenience. Specific implementation details are left out for the sake of clarity, but the source code is available at the URL mentioned in the main manuscript (Abstract→Availability).

The core of `FastCMH` is the routine `get_testable_regions`, which aims at computing Tarone's adjusted significance threshold δ_{tar} and retrieving the set of testable genomic regions $\mathcal{R}_T(\delta_{tar})$. In this section, we describe precisely how specific steps of `get_testable_regions` can be implemented efficiently.

Algorithm 1 `FastCMH`

Input: Dataset $\mathcal{G} = \{\mathbf{g}_i, y_i, c_i\}_{i=1}^n$, desired FWER α
Output: Set of non-overlapping conditionally associated genomic regions $\mathcal{R}_{sig,filt} = \{[\![t_s, t_e]\!] \,|\, p([\![t_s, t_e]\!]) \leq \delta_{tar}\}$

1: $(\delta_{tar}, \mathcal{R}_T(\delta_{tar})) \leftarrow$ `get_testable_regions`$(\mathcal{G}, \alpha)$
2: $\mathcal{R}_{sig,raw} \leftarrow \{[\![t_s, t_e]\!] \in \mathcal{R}_T(\delta_{tar}) \,|\, p([\![t_s, t_e]\!]) \leq \delta_{tar}\}$
3: $\mathcal{R}_{sig,filt} \leftarrow$ `filter_overlapping_regions`$(\mathcal{R}_{sig,raw})$
4: Return $\mathcal{R}_{sig,filt}$

Algorithm 2 `get_testable_regions`

Input: Dataset $\mathcal{G} = \{\mathbf{g}_i, y_i, c_i\}_{i=1}^n$, desired FWER α
Output: Tarone's adjusted significance threshold δ_{tar} and set of testable genomic regions $\mathcal{R}_T(\delta_{tar})$

1: $\delta \leftarrow 1$, $\mathcal{R}_T(\delta) \leftarrow \emptyset$
2: $\mathcal{R}_{cand} \leftarrow \{[\![t_s, t_e]\!] \,|\, 1 \leq t_s \leq t_e \leq l\}$
3: **for** $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$ **do** ▷ Regions in $\mathcal{R}_{cand}$ enumerated firstly in increasing order of length and then starting position
4: **if** $p_{min}([\![t_s, t_e]\!]) \leq \delta$ **then**
5: $\mathcal{R}_T(\delta) \leftarrow \mathcal{R}_T(\delta) \cup \{[\![t_s, t_e]\!]\}$
6: **while** $\delta > \alpha/|\mathcal{R}_T(\delta)|$ **do**
7: Decrease δ
8: $\mathcal{P} \leftarrow \{[\![t_s, t_e]\!] \in \mathcal{R}_T(\delta) \,|\, p_{min}([\![t_s, t_e]\!]) > \delta\}$
9: $\mathcal{R}_T(\delta) \leftarrow \mathcal{R}_T(\delta) \setminus \mathcal{P}$
10: **if** `pruning_condition`$([\![t_s, t_e]\!])$ **then**
11: Remove all $[\![t'_s, t'_e]\!] \supset [\![t_s, t_e]\!]$ from $\mathcal{R}_{cand}$
12: Return $\delta_{tar} \leftarrow \delta$ and $\mathcal{R}_T(\delta_{tar}) = \mathcal{R}_T(\delta)$

In Line 4 of Algorithm 2, we assess if the genomic region $[\![t_s, t_e]\!]$ being processed is testable. This requires computing the corresponding meta-marker $g_i([\![t_s, t_e]\!])$ for all individuals i in the dataset. If implemented naively, this would have complexity $O(n(t_e - t_s + 1))$ with $n = \sum_{h=1}^k n_h$ being the total number of individuals and $t_e - t_s + 1$ the number of variants in the genomic region. However, since candidate genomic regions are enumerated in increasing order of length, it is possible to obtain the meta-marker $g_i([\![t_s, t_e]\!])$ corresponding to region $[\![t_s, t_e]\!]$ from either the meta-marker $g_i([\![t_s, t_e - 1]\!])$ of region $[\![t_s, t_e - 1]\!]$ or the meta-marker $g_i([\![t_s + 1, t_e]\!])$ of region $[\![t_s + 1, t_e]\!]$. Therefore, the complexity can be reduced to $O(n)$. Moreover, since $g_i([\![t_s, t_e - 1]\!]) = 1$ or $g_i([\![t_s + 1, t_e]\!]) = 1$ imply that $g_i([\![t_s, t_e]\!]) = 1$, the complexity can be further reduced to $O(n - x_{prev})$, where x_{prev} is the number of individuals that have meta-marker equal to 1 in either region $[\![t_s, t_e - 1]\!]$ or region $[\![t_s + 1, t_e]\!]$. Implementing the computation of the meta-marker in this manner in Line 4 of Algorithm 2 greatly increases the efficiency of the algorithm with only a moderate increase in memory usage, equivalent to storing a second copy of the original dataset in memory.

In Line 7 of Algorithm 2, the tentative significance threshold δ is decreased until the FWER condition $\delta \leq \alpha/|\mathcal{R}_T(\delta)|$ is satisfied again. In practice, we implement this step as $\delta \leftarrow 10^{-\Delta}\delta$, where Δ is an implementation-dependent hyperparameter. This is equivalent to performing grid-search on δ, with logarithmically-spaced candidate values with step-size Δ in the log scale. Provided that Δ is not too large, we found this hyperparameter to have a negligible effect on the results. We fixed $\Delta = 0.06$ throughout our experiments, corresponding to considering 500 values of δ in a logarithmic grid between $\delta = 1$ and $\delta = 10^{-30}$.

In Lines 8 and 9, we remove all previously enumerated genomic regions that were found to be testable using previous values of δ, but are no longer testable due to δ having been decreased in Line 7 of Algorithm 2. Based

on the scheme to decrease δ described in the paragraph above, the set $\mathcal{P}$ of genomic regions to be removed from $\mathcal{R}_T(\delta)$ is composed of those genomic regions $[\![t_s, t_e]\!]$ currently in $\mathcal{R}_T(\delta)$ that satisfy $\delta < p_{min}([\![t_s, t_e]\!]) \leq 10^{\Delta}\delta$. This property shows that it is possible to implement Lines 8 and 9 with $O(1)$ complexity by using a data-structure for $\mathcal{R}_T(\delta)$ that stores genomic regions in different bins according to their minimum attainable p-value. More precisely, we assign a genomic region $[\![t_s, t_e]\!]$ to the i-th bin if $10^{-i\Delta} < p_{min}([\![t_s, t_e]\!]) \leq 10^{-(i-1)\Delta}$. With this data -structure, each execution of Lines 8 and 9 corresponds to removing exactly one bin from $\mathcal{R}_T(\delta)$, a $O(1)$ operation that requires no search.

Finally, a possibility that should be considered is the case in which $\mathcal{R}_T(\delta)$ is too large to fit in memory. While this has not occurred in any of our experiments, it could be the case if `FastCMH` was applied to biobank-sized datasets. Notice that, in principle, in order to compute Tarone's adjusted significance threshold δ_{tar}, only the number of testable genomic regions $|\mathcal{R}_T(\delta)|$ is needed. In other words, each bin described in the previous paragraph needs to store only the number of elements currently in the bin, not the elements themselves. Once δ_{tar} has been obtained, the enumeration procedure can be executed again from scratch but with $\delta = \delta_{tar}$ fixed throughout the enumeration. During this second enumeration, for each genomic region with $p_{min}([\![t_s, t_e]\!]) \leq \delta_{tar}$, the p-value for the genomic region can be computed to assess significance. Implementing `FastCMH` in this manner avoids the need to store $\mathcal{R}_T(\delta)$ in memory, only at the cost of roughly doubling the runtime, due to the need to execute the enumeration procedure twice.

S1.1.1 Filtering procedure used in `FastCMH`

Algorithm 3 describes the `filter_overlapping_regions` method used in Step 3 in Algorithm 1 above.

Algorithm 3 `filter_overlapping_regions`

Input: Set of regions with the associated p-values, $\mathcal{R}_{sig,raw} = \{([\![t_s, t_e]\!], p([\![t_s, t_e]\!]))\}$
Output: Set of non-overlapping regions $\mathcal{R}_{sig,filt} \subset \mathcal{R}_{sig,raw}$ which are most significant in each cluster

1: Determine disjoint clusters $C_1, C_2, \ldots, C_\ell$, where each cluster is the union of a subset of $\mathcal{R}_{sig,raw}$
2: **for** each $[\![t_s, t_e]\!] \in \mathcal{R}_{sig,raw}$ **do**
3: **for** $j \in \{1, 2, \ldots, \ell\}$ **do**
4: **if** $[\![t_s, t_e]\!]$ belongs to C_j **then**
5: Assign $[\![t_s, t_e]\!]$ the label j
6: **for** $j \in \{1, 2, \ldots, \ell\}$ **do**
7: Find the region $[\![t_{s,j}, t_{e,j}]\!]$ that has the smallest p-value $p([\![t_{s,j}, t_{e,j}]\!])$ amongst all region in C_j
8: Let $\mathcal{R}_{sig,filt} = \{[\![t_{s,j}, t_{e,j}]\!] \,|\, j = 1, 2, \ldots, \ell\}$
9: Return $\mathcal{R}_{sig,filt}$

In simple terms, the regions in $\mathcal{R}_{sig,raw}$ are first grouped into *clusters*; if one considers the union of all regions in $\mathcal{R}_{sig,raw}$, then there would be several groups of overlapping regions which each form larger contiguous regions. We call each of these larger contiguous regions a *cluster*. Note that the clusters do not overlap, but the regions within each cluster do overlap. Figure S1 shows an example of a cluster containing four regions. The clusters can be determined in Line 1 by following two rules: (i) every regions must belong to one cluster, and (ii) if two regions overlap, they belong to the same cluster.

After the clusters $C_1, C_2, \ldots, C_\ell$ have been determined, another pass is made through the regions and each region is given the label $\{1, 2, \ldots, \ell\}$ of the cluster to which it belongs (Lines 2 to 5).

Next, the region in each cluster which has the smallest p-value is determined (Lines 6 to 7). In the case of ties (two or more region with the same minimum p-value), the region that has the longest length is returned. If the lengths are the same, then the region with the smallest τ is returned.

Finally, we construct $\mathcal{R}_{sig,filt}$ in Line 8 as the collection of those regions that are the most significant in each cluster (i.e. one interval per cluster). Figure S1 illustrates the procedure for a single cluster; it illustrates how four overlapping regions (red) form a single cluster (magenta), and the filtering process identifies the region with the smallest p-value (the green region, p=1e-9). This filtering procedure was first used in Llinares-López *et al.* (2015).

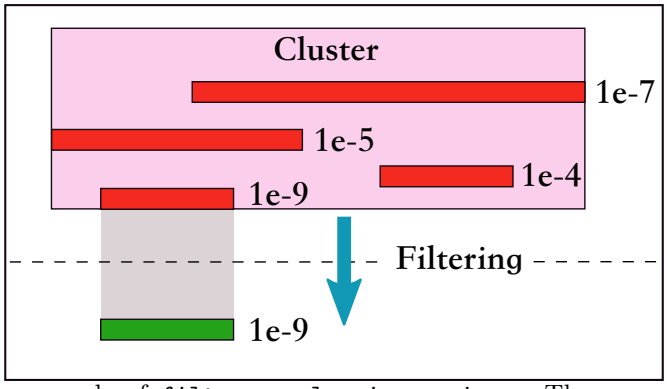

Figure S1: An illustrative example of `filter_overlapping_regions`. There are four overlapping significant regions (in red), which together form a single cluster (magenta). The result of the filtering in this cluster is the green region, since it has the smallest p-value, p=1e-9

S1.2 Theoretical aspects of `FastCMH`

S1.2.1 p-value for the CMH test

We use the same notation as in the contingency table in the main document. n_h is the number of individuals with $c_i = h$, divided into $n_{1,h}$ cases and $n_{2,h}$ controls. Similarly, x_h is the number of individuals with $c_i = h$ for which the meta-marker $g_i([\![t_s, t_e]\!])$ takes value 1, a_h of which are cases and $x_h - a_h$ controls. Using those parameters $\{n_h, n_{1,h}, x_h, a_h\}_{h=1}^k$, we can compute the p-value $p([\![t_s, t_e]\!])$ for a genomic region $[\![t_s, t_e]\!]$ under the CMH test as:

$$p([\![t_s, t_e]\!]) = 1 - F_{\chi^2_1}\left(\frac{\left(\sum_{h=1}^k a_h - \frac{x_h n_{1,h}}{n_h}\right)^2}{\sum_{h=1}^k \frac{n_{1,h}}{n_h}\left(1 - \frac{n_{1,h}}{n_h}\right) x_h \left(1 - \frac{x_h}{n_h}\right)}\right)$$

with $F_{\chi^2_1}(\cdot)$ being the cumulative distribution function of a χ^2 random variable with 1 degree of freedom.

S1.2.2 Minimum attainable p-value for the CMH test

Let $[\![t_s, t_e]\!]$ be an arbitrary genomic region with table margins $\{x_h, n_{1,h}, n_h\}_{h=1}^k$. Then, as shown in Papaxanthos *et al.* (2016), Tarone's minimum attainable p-value $p_{min}([\![t_s, t_e]\!])$ when using the CMH test can be obtained as:

$$p_{min}([\![t_s, t_e]\!]) = p_{min}(x_1, \ldots, x_k) = 1 - F_{\chi^2_1}\left(T_{max}(x_1, \ldots, x_k)\right)$$

where $T_{max}(x_1, \ldots, x_k) = \max(T(a_{min}, x_1, \ldots, x_k), T(a_{max}, x_1, \ldots, x_k))$, $a_{min} = \sum_{h=1}^k \max(0, x_h - n_{2,h})$, $a_{max} = \sum_{h=1}^k \min(x_h, n_{1,h})$ and:

$$T(a, x_1, \ldots, x_k) = \frac{\left(a - \sum_{h=1}^k x_h \frac{n_{1,h}}{n_h}\right)^2}{\sum_{h=1}^k \frac{n_{1,h}}{n_h}\left(1 - \frac{n_{1,h}}{n_h}\right) x_h \left(1 - \frac{x_h}{n_h}\right)}$$

Since all genomic regions $[\![t_s, t_e]\!]$ have the same values for $\{n_{1,h}, n_h\}_{h=1}^k$, they can be treated as constants, and the dependence of $p_{min}([\![t_s, t_e]\!]) = p_{min}(x_1, \ldots, x_k)$ on them has been omitted to simplify the notation.

S1.2.3 Pruning condition for the CMH test

In this section, we revise the novel pruning condition for the CMH test proposed in Papaxanthos *et al.* (2016), which we employ in Line 10 of Algorithm 2 as discussed in the main manuscript. For the sake of clarity, we adapt their notation and rephrase their statements to match our setting: mining significantly associated genomic regions while correcting for a categorical covariate. For a more in-depth treatment of the theory, we refer the reader to Papaxanthos *et al.* (2016).

As discussed in Section 3.2 of the main manuscript, the pruning condition involves computing a lower bound $\tilde{p}_{min}$ on the minimum attainable p-value p_{min}. This lower bound $\tilde{p}_{min}$ is defined only on the set of genomic regions $\mathcal{R}_C = \{[\![t_s, t_e]\!] \,|\, x_h \in [\![\max(n_{1,h}, n_h - n_{1,h}), n_h]\!] \; \forall\, h \in \{1, \ldots, k\}\}$, where x_h is the number of individuals belonging to category h of the covariate for which the meta-marker of region $[\![t_s, t_e]\!]$ has value 1. Pruning never occurs for genomic regions which do not belong to $\mathcal{R}_C$. In practice, a large proportion of all candidate genomic regions

belong to $\mathcal{R}_C$ because their corresponding genomic meta-markers accumulate meta-minor alleles as the length of the region grows.

For convenience, we introduce the following notation in the remainder of this section. Given a genomic region $[\![t_s, t_e]\!]$, we define $\mathcal{R}_{sup}([\![t_s, t_e]\!])$ to be the set of all *genomic super-regions* $[\![t'_s, t'_e]\!]$ which contain $[\![t_s, t_e]\!]$, i.e. $\mathcal{R}_{sup}([\![t_s, t_e]\!]) = \{[\![t'_s, t'_e]\!] \,|\, [\![t'_s, t'_e]\!] \supseteq [\![t_s, t_e]\!]\}$. Also, for a genomic region $[\![t_s, t_e]\!]$, we define its vector of meta-minor allele counts $\mathbf{x} \in \mathbb{N}^k$ as $\mathbf{x} = (x_1, x_2, \ldots, x_k)$. Whenever relevant, we make explicit the dependence of the minimum attainable p-value of a genomic region $[\![t_s, t_e]\!]$ on $\mathbf{x}$ by denoting $p_{min}([\![t_s, t_e]\!])$ as $p_{min}(\mathbf{x})$. Finally, we write $\mathbf{x}' \geq \mathbf{x}$ if $\mathbf{x}'_h \geq \mathbf{x}_h \; \forall \, h = 1, \ldots, k$.

Definition 1. *Let $[\![t_s, t_e]\!] \in \mathcal{R}_C$ be a genomic region and let $\mathbf{x}$ be its k-dimensional vector of meta-minor allele counts. The lower bound $\tilde{p}_{min} : [\![t_s, t_e]\!] \in \mathcal{R}_C \rightarrow \tilde{p}_{min}([\![t_s, t_e]\!])$ is defined as:*

$$\begin{aligned} \tilde{p}_{min}([\![t_s, t_e]\!]) &= \min_{[\![t'_s, t'_e]\!] \in \mathcal{R}_{sup}([\![t_s, t_e]\!])} p_{min}([\![t'_s, t'_e]\!]) \\ &= \min_{\mathbf{x}' \geq \mathbf{x}} p_{min}(\mathbf{x}') \end{aligned} \tag{1}$$

The lower bound $\tilde{p}_{min}$ satisfies the following two properties, which together allow efficiently pruning the search space:

Property 1. *$p_{min}([\![t_s, t_e]\!]) \geq \tilde{p}_{min}([\![t_s, t_e]\!])$ holds for all genomic regions $[\![t_s, t_e]\!] \in \mathcal{R}_C$.*

Proof. The proof follows directly from the definition of $\tilde{p}_{min}$. Indeed, $\tilde{p}_{min}([\![t_s, t_e]\!]) = \min_{[\![t'_s, t'_e]\!] \in \mathcal{R}_{sup}([\![t_s, t_e]\!])} p_{min}([\![t'_s, t'_e]\!])$ implies $\tilde{p}_{min}([\![t_s, t_e]\!]) \leq p_{min}([\![t_s, t_e]\!])$ since $[\![t_s, t_e]\!] \in \mathcal{R}_{sup}([\![t_s, t_e]\!])$. □

Property 2. *$\tilde{p}_{min}([\![t'_s, t'_e]\!]) \geq \tilde{p}_{min}([\![t_s, t_e]\!])$ holds for all genomic regions $[\![t_s, t_e]\!] \in \mathcal{R}_C$, $[\![t'_s, t'_e]\!] \in \mathcal{R}_{sup}([\![t_s, t_e]\!])$.*

Proof. $\tilde{p}_{min}([\![t_s, t_e]\!]) \leq \tilde{p}_{min}([\![t'_s, t'_e]\!])$ follows from the definition of $\tilde{p}_{min}$ and from $\mathcal{R}_{sup}([\![t'_s, t'_e]\!]) \subset \mathcal{R}_{sup}([\![t_s, t_e]\!])$. □

These properties lead to the main statement: $\tilde{p}_{min}$ lets us prune a large proportion of non-testable candidate regions without actually computing their minimum attainable p-values p_{min}.

Theorem 1 (Papaxanthos *et al.* (2016), Theorem 1)**.** *Let δ be the current adjusted significance threshold. If $\tilde{p}_{min}([\![t_s, t_e]\!]) > \delta$ holds for $[\![t_s, t_e]\!] \in \mathcal{R}_C$, then all genomic regions $[\![t'_s, t'_e]\!] \in \mathcal{R}_{sup}([\![t_s, t_e]\!])$ are not testable and can be pruned from the search space $\mathcal{R}_{cand}$.*

Proof. Theorem 1 follows from the definition of $\tilde{p}_{min}$ combined with Properties 1 and 2, indeed: $\tilde{p}_{min}([\![t_s, t_e]\!]) > \delta \Rightarrow \forall [\![t'_s, t'_e]\!] \in \mathcal{R}_{sup}([\![t_s, t_e]\!]),\; p_{min}([\![t'_s, t'_e]\!]) \underset{Prop.\ 1}{\geq} \tilde{p}_{min}([\![t'_s, t'_e]\!]) \underset{Prop.\ 2}{\geq} \tilde{p}_{min}([\![t_s, t_e]\!]) > \delta$. Therefore, it ensues that $[\![t'_s, t'_e]\!]$ is not-testable. □

Theorem 1 explains why the `pruning_condition` in Line 10 of Algorithm 2 evaluates to True if and only if $\tilde{p}_{min}([\![t_s, t_e]\!]) > \delta$. In this case, all *genomic super-regions* of $[\![t_s, t_e]\!]$ cannot be testable and are pruned from the set of candidate regions $\mathcal{R}_{cand}$. Property 2 is key to the pruning step described in Theorem 1, as it is that *monotonicity property* what allows discarding all super-regions of a genomic region. It also explains the necessity of using a lower bound $\tilde{p}_{min}$ instead of the minimum p-value p_{min}, which is not monotonic in $\mathcal{R}_C$ when $k \geq 2$. This problem does *not* occur when $k = 1$, i.e. when categorical covariates are not taken into account.

Since the pruning criterion has to be evaluated for all enumerated regions, it is essential to compute $\tilde{p}_{min}$ efficiently. However, naively evaluating $\tilde{p}_{min}$, as defined in Equation 1, would be too computationally intensive as it would require one to optimize p_{min} over a set of size $\prod_{h=1}^{k} \min(n_{1,h}, n_h - n_{1,h}) = O(m^k)$ with $m = (\prod_{h=1}^{k} \min(n_{1,h}, n_h - n_{1,h}))^{\frac{1}{k}}$. However, in Papaxanthos *et al.* (2016) the authors proved that an efficient algorithm to compute $\tilde{p}_{min}([\![t_s, t_e]\!]) \; \forall \, [\![t_s, t_e]\!] \in \mathcal{R}_C$ exists:

Theorem 2 (Papaxanthos *et al.* (2016), Theorem 2)**.** *Let $[\![t_s, t_e]\!] \in \mathcal{R}_C$ be a genomic region and let $\mathbf{x}$ be its k-dimensional vector of meta-minor allele counts, with k being the number of states of the categorical covariate. Define the quantities $\beta_h^l = \frac{n_{1,h}}{n_h} \frac{x_h}{n_h}$, $\beta_h^r = \left(1 - \frac{n_{1,h}}{n_h}\right) \frac{x_h}{n_h}$ for each $h = 1, \ldots, k$ and let π_l and π_r be permutations $\pi_l, \pi_r : [\![1, k]\!] \mapsto [\![1, k]\!]$ that order the respective sets $\left\{\beta_h^l\right\}_{h=1}^{k}$ and $\{\beta_h^r\}_{h=1}^{k}$ in increasing order. Then, the solution $\mathbf{x}^*$ of the optimization problem $\mathbf{x}^* = \underset{\mathbf{x}' \geq \mathbf{x}}{\operatorname{argmin}} \, p_{min}(\mathbf{x}')$ defining $\tilde{p}_{min}([\![t_s, t_e]\!])$ satisfies one of the following two conditions:*

1. *$\mathbf{x}^*_{\pi_l(j)} = \mathbf{x}_{\pi_l(j)}$ for $j \leq \kappa$ and $\mathbf{x}^*_{\pi_l(j)} = 0$ for $j > \kappa$, $\kappa \in [\![1, k]\!]$*

2. $\mathbf{x}^*_{\pi_r(j)} = \mathbf{x}_{\pi_r(j)}$ *for* $j \leq \kappa$ *and* $\mathbf{x}^*_{\pi_r(j)} = 0$ *for* $j > \kappa$, $\kappa \in [\![1, k]\!]$

In a nutshell, Theorem 2 states that only $2k$ different points $\mathbf{x}'$, k of them satisfying condition 1 and k satisfying condition 2, could be the solution to the optimization problem $\mathbf{x}^* = \underset{\mathbf{x}' \geq \mathbf{x}}{\operatorname{argmin}} p_{min}(\mathbf{x}')$, which defines the lower bound $\tilde{p}_{min}([\![t_s, t_e]\!])$ needed to evaluate the pruning criterion given by Theorem 1. By evaluating the minimum attainable p-value function for the CMH test, $p_{min}(\mathbf{x}')$, at only those $2k$ points, the solution to the optimization problem can be found efficiently.

According to Condition 1, k possible candidates $\mathbf{x}'$ to be the solution can be obtained as follows. Firstly, compute $\beta^l_h = \frac{n_{1,h}}{n_h}\frac{x_h}{n_h}$ for each category $h = 1, \ldots, k$. Next, let $\pi_l : [\![1, k]\!] \mapsto [\![1, k]\!]$ be a permutation that sorts those values in ascending order, that is, $\beta^l_{\pi_l(1)} \leq \beta^l_{\pi_l(2)} \cdots \beta^l_{\pi_l(k-1)} \leq \beta^l_{\pi_l(k)}$. Then, the first solution candidate, corresponding to $\kappa = 1$ in the statement of Condition 1 in Theorem 2, is a vector $\mathbf{x}'$ such that its $\pi_l(1)$ entry has value $\mathbf{x}_{\pi_l(1)}$, with all other entries having value 0. The next solution candidate, corresponding to $\kappa = 2$, is another vector $\mathbf{x}'$ such that its $\pi_l(1)$ entry has value $\mathbf{x}_{\pi_l(1)}$ and its entry $\pi_l(2)$ has value $\mathbf{x}_{\pi_l(2)}$, with all other entries having value 0. Eventually, the last candidate, corresponding to $\kappa = k$, is $\mathbf{x}' = \mathbf{x}$.

The k solution candidates corresponding to Condition 2 are obtained analogously, with the sole difference that the permutation $\pi_l : [\![1, k]\!] \mapsto [\![1, k]\!]$ is obtained by sorting the values $\beta^r_h = \left(1 - \frac{n_{1,h}}{n_h}\right)\frac{x_h}{n_h}$, $h = 1, \ldots, k$ in ascending order.

Therefore, Theorem 2 leads to an algorithm to evaluate the pruning condition in only $O(k \log k)$ time, since the sorting steps dominate the total runtime complexity. That method, which we use in the core routine `get_testable_regions` of `FastCMH`, is reproduced in Algorithm 4 for convenience, adapting the original notation in Papaxanthos *et al.* (2016) to our setting.

Algorithm 4 `pruning_condition`

Input: Table margins $\{x_h, n_{1,h}, n_h\}_{h=1}^k$ for genomic region $[\![t_s, t_e]\!]$, current adjusted significance threshold δ
Output: A Boolean (`true` or `false`) indicating if the pruning condition applies to genomic region $[\![t_s, t_e]\!]$

1: **if** $[\![t_s, t_e]\!] \notin \mathcal{R}_C$ **then**
2: Return `false`
3: $\beta^l_h \leftarrow \frac{n_{1,h}}{n_h}\frac{x_h}{n_h}$, $\beta^r_h \leftarrow \left(1 - \frac{n_{1,h}}{n_h}\right)\frac{x_h}{n_h}$ for each $h = 1, \ldots, k$
4: $\pi_l \leftarrow$ `argsort`$(\{\beta^l_h\}_{h=1}^k)$, $\pi_r \leftarrow$ `argsort`$(\{\beta^r_h\}_{h=1}^k)$ ▷ Sort sets in ascending order
5: $f_l \leftarrow 0$, $f_r \leftarrow 0$
6: $g \leftarrow 0$, $g_r \leftarrow 0$
7: **for** $i \in [\![1, k]\!]$ **do**
8: $f_l \leftarrow f_l + \left(1 - \frac{n_{1,\pi_l(i)}}{n_{\pi_l(i)}}\right)(n_{\pi_l(i)} - x_{\pi_l(i)})$, $f_r \leftarrow f_r + \frac{n_{1,\pi_r(i)}}{n_{\pi_r(i)}}(n_{\pi_r(i)} - x_{\pi_r(i)})$
9: $g_l \leftarrow g_l + \frac{n_{1,\pi_l(i)}}{n_{\pi_l(i)}}\left(1 - \frac{n_{1,\pi_l(i)}}{n_{\pi_l(i)}}\right)x_{\pi_l(i)}\left(1 - \frac{x_{\pi_l(i)}}{n_{\pi_l(i)}}\right)$, $g_r \leftarrow g_r + \frac{n_{1,\pi_r(i)}}{n_{\pi_r(i)}}\left(1 - \frac{n_{1,\pi_r(i)}}{n_{\pi_r(i)}}\right)x_{\pi_r(i)}\left(1 - \frac{x_{\pi_r(i)}}{n_{\pi_r(i)}}\right)$
10: $p^l_i \leftarrow 1 - F_{\chi^2}\left(\frac{f_l^2}{g_l}\right)$, $p^r_i \leftarrow 1 - F_{\chi^2}\left(\frac{f_r^2}{g_r}\right)$
11: $\tilde{p}_{min}([\![t_s, t_e]\!]) \leftarrow \min(p^l_1, \ldots, p^l_k, p^r_1, \ldots, p^r_k)$
12: **if** $\tilde{p}_{min}([\![t_s, t_e]\!]) > \delta$ **then**
13: Return `true`
14: **else**
15: Return `false`

Firstly, in Line 1, the algorithm checks if the current genomic region $[\![t_s, t_e]\!]$ belongs to the set $\mathcal{R}_C$ defined earlier in this section. If not, the pruning condition does not apply to the genomic region currently being processed and the algorithm terminates. Otherwise, the procedure described by Theorem 2 begins. In Line 3, we evaluate the quantities β^l_h and β^r_h for all tables $h = 1, \ldots, k$, a step with complexity $O(k)$. Next, in Line 4, the sets $\{\beta^l_h\}_{h=1}^k$ and $\{\beta^r_h\}_{h=1}^k$ are sorted to obtain the permutations π_l, π_r described in the theorem. This is the step that dominates the overall complexity, requiring $O(k \log k)$ time. According to conditions 1. and 2. in Theorem 2, each of the two permutations π_l, π_r implicitly defines k candidate values $\mathbf{x}'$ to solve the optimization problem $\mathbf{x}^* = \underset{\mathbf{x}' \geq \mathbf{x}}{\operatorname{argmin}} p_{min}(\mathbf{x}')$ that defines the lower bound $\tilde{p}_{min}([\![t_s, t_e]\!])$. Between Lines 5 and 10, the algorithm evaluates $p_{min}(\mathbf{x}')$ for each of those $2k$ candidate values, storing them in the variables $p^l_1, \ldots, p^l_k, p^r_1, \ldots, p^r_k$. Notice that, since the values are computed incrementally, the complexity of this step is $O(k)$. Finally, in Line 11, the value for the lower bound $\tilde{p}_{min}([\![t_s, t_e]\!])$ is obtained by taking the minimum value of $p_{min}(\mathbf{x}')$ among the $2k$ candidates

previously computed, requiring also $O(k)$ time. Using that value, the pruning condition $\tilde{p}_{min}([\![t_s, t_e]\!]) > \delta$ given by Theorem 1 can be finally evaluated.

Removal of pruned candidates from $\mathcal{R}_{cand}$ An example illustrating how to perform the pruning of the search space in Line 10 of Algorithm 2 is shown in Figure S2. Assuming there are only 6 markers, the first level in the figure (top row) shows all regions of length 1, i.e. each marker by itself. In the second level we find all regions of length 2, and similarly for the other levels.

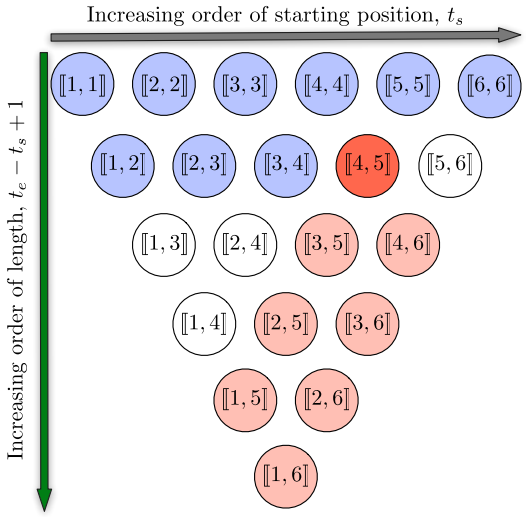

Figure S2: Schematic illustration of how pruning can efficiently reduce the number of candidate genomic regions. In this case, the pruning condition applies to the region $[\![4,5]\!]$ (dark red) and all regions which contain it (in light red).

Let us assume that the candidate region that is being evaluated by Algorithm 2 in Line 10 is $[\![t_s, t_e]\!] = [\![4,5]\!]$ (marked in dark red in Figure S2). If the function `pruning_condition`($[\![4,5]\!]$) returns true, then all regions that contain $[\![4,5]\!]$ can be pruned from the search space. Line 10 then prunes the regions marked in light red in Figure S2.

S1.3 Generalization of the meta-marker

In Section 2.1 of the main manuscript, we considered a setting in which the genotype of an individual i is represented as an ordered sequence of l binary genomic markers, $\mathbf{g}_i = (\mathbf{g}_i[1], \mathbf{g}_i[2] \ldots, \mathbf{g}_i[l])$ with $\mathbf{g}_i[t] \in \{0,1\}$. The meta-marker for a genomic region $[\![t_s, t_e]\!]$ was defined as $g_i([\![t_s, t_e]\!]) = \max(\mathbf{g}_i[t_s], \mathbf{g}_i[t_s+1], \ldots, \mathbf{g}_i[t_e])$. Intuitively, if the binary genotypes are encoded such that $\mathbf{g}_i[t]$ indicates the presence of at least one minor allele at position t (i.e. a dominant encoding), then the meta-marker of a genomic region can be interpreted as an indicator of the presence or absence of minor alleles in the genomic region. Alternatively, from an statistical point of view, combining the markers within a genomic region using the max operator offers the highest sensitivity to aggregate weak signals, but is also the most affected by noisy, irrelevant variants included in the region. Both biologically and statistically, it might be interesting to generalize our working definition of meta-marker to offer greater flexibility. As we show next, this can be done with no additional changes to `FastCMH`, other than the computation of the meta-marker itself.

Consider an ordered sequence of l genomic markers, $\mathbf{g}_i = (\mathbf{g}_i[1], \mathbf{g}_i[2] \ldots, \mathbf{g}_i[l])$ with $\mathbf{g}_i[t] \in \{0,1,2\}$, corresponding to an additive encoding of the markers. We can generalize the definition of meta-marker as:

$$g_i([\![t_s, t_e]\!]; m_g) = \left[\sum_{t=t_s}^{t_e} \mathbf{g}_i[t] \geq m_g \right] \tag{2}$$

where $m_g \in \mathbb{N}$ is a fixed allelic burden threshold and $[\cdot] = 1$ if its argument is true, and $[\cdot] = 0$ otherwise. This corresponds to defining the meta-marker of a region to be 1 if it contains at least m_g minor alleles. Thus, the definition of meta-marker used throughout the main manuscript corresponds to the special case $m_g = 1$.

All properties of the meta-marker necessary for Definition 1 and Theorems 1 and 2 to hold are satisfied regardless of the value of $m_g \in \mathbb{N}$. Therefore, `FastCMH` can be straightforwardly generalized to arbitrary m_g with no additional modifications.

If multiple values of $m_g \in \mathcal{M}_g$ are to be considered, a naive approach would be to run `FastCMH` independently for each value, dividing the individual adjusted significance threshold δ_{tar} obtained with `FastCMH` for each value of m_g by the total number of values considered $|\mathcal{M}_g|$ before assessing the significance of each genomic region. However, this would be computationally inefficient, as the entire enumeration process would need to be repeated $|\mathcal{M}_g|$ times.

An important property we can exploit to solve that limitation is that, for $m'_g \geq m_g$, we have $g_i([\![t_s, t_e]\!]; m'_g) = 1 \Rightarrow g_i([\![t_s, t_e]\!]; m_g) = 1$. In turn, this implies $\mathbf{x}(m_g) \geq \mathbf{x}(m'_g)$, where $\mathbf{x}(m_g)$ and $\mathbf{x}(m'_g)$ are the vectors of meta-allele counts obtained for genomic region $[\![t_s, t_e]\!]$ when using values m_g and m'_g in the definition of the meta-marker. Since the table margins $\{n_{1,h}, n_h\}_{h=1}^{k}$ do not depend on m_g, this implies that $\tilde{p}_{min}([\![t_s, t_e]\!]; m'_g) \leq \tilde{p}_{min}([\![t_s, t_e]\!]; m_g)$ for all genomic regions $[\![t_s, t_e]\!] \in \mathcal{R}_C$. As a consequence, we obtain the following key result:

Proposition 1. *If the pruning condition defined by Theorem 1 applies to a genomic region $[\![t_s, t_e]\!] \in \mathcal{R}_C$ using a generalized meta-marker with allelic burden threshold m'_g, then the pruning condition will also apply for the same region if the generalized meta-marker is defined with any other allelic burden threshold satisfying $m_g \leq m'_g$.*

Proposition 1 shows that, to run `FastCMH` with multiple values of $m_g \in \mathcal{M}_g$, it is possible to process all values at once, performing search space pruning according to the value of the meta-marker for the largest allelic burden threshold in $\mathcal{M}_g$.

S1.4 Extending `FastCMH` to use False Discovery Rate (FDR) control

Following a suggestion from a reviewer, we incorporated a False Discovery Rate (FDR) procedure that uses Tarone's method into `FastCMH`, and we call this modified procedure `FastCMH-FDR`. This section briefly discusses this modification. The results comparing `FastCMH` and `FastCMH-FDR`, obtained on different datasets, can be found in Section S3.

We note that the paper Gilbert (2005) was the first to combine Tarone's method (Tarone, 1990) with the False Discovery Rate (FDR) procedure (Benjamini and Hochberg, 1995a), and this is the method upon which `FastCMH-FDR` is based. On the other hand, `FastCMH` uses Tarone's procedure which controls the Family-wise Error Rate. We start with a brief review of the standard FDR procedure from Benjamini and Hochberg (1995a).

Original False Discovery Rate (FDR) procedure

Suppose there are m null hypotheses $H_1, H_2, \ldots, H_m$ being tested simulataneously. Define:

- $\mathbf{R}$ to be the number of null hypotheses rejected by a particular testing procedure,
- $\mathbf{V}$ to be the number of *true* null hypotheses rejected by a particular testing procedure.

The *false discovery rate* (FDR) is then defined as:

$$\text{FDR} = \begin{cases} \mathrm{E}\left[\mathbf{V}/\mathbf{R}\right], & \text{if } \mathbf{R} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Every hypothesis test H_i results in a p-value p_i. In order to compute the FDR, one orders first the m p-values:

$$p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$$

so that $p_{(i)}$ is the ith largest p-value, and $H_{(i)}$ is the corresponding null hypothesis. Next, for a pre-specified significance threshold $\alpha \in (0, 1)$, one checks in turn:

$$\begin{aligned} p_{(m)} &\leq \alpha, \\ p_{(m-1)} &\leq \left(\frac{m-1}{m}\right)\alpha, \\ &\vdots \\ p_{(i)} &\leq \left(\frac{i}{m}\right)\alpha, \\ &\vdots \\ p_{(2)} &\leq \left(\frac{2}{m}\right)\alpha, \\ p_{(1)} &\leq \left(\frac{1}{m}\right)\alpha, \end{aligned}$$

stopping at the first (and largest) index k such that

$$p_{(k)} \leq \left(\frac{k}{m}\right)\alpha.$$

Then, once k is found, the procedure specifies rejecting the null hypotheses $H_{(1)}, H_{(2)}, \ldots, H_{(k)}$. So, in other words, we start from the LARGEST p-value, and work down to the smallest p-value.

In Benjamini and Hochberg (1995a) it was shown that this procedure controls the FDR at significance level α (i.e. FDR $\leq \alpha$) for independent, *continuous* test statistics.

In Benjamini and Yekutieli (2001) it was shown that this procedure also controls FDR $\leq \alpha$ for independent, *discrete* test statistics.

In Benjamini and Yekutieli (2001) it was also shown that if, instead of checking

$$p_{(i)} \leq \left(\frac{i}{m}\right)\alpha,$$

one rather checks

$$p_{(i)} \leq \left(\frac{i}{m}\right)\widetilde{\alpha}, \qquad \widetilde{\alpha} = \frac{\alpha}{\sum_{j=1}^{m}(1/j)},$$

then this procedure also controls the FDR for *dependent* test statistics, under certain kinds of dependence structures (known as *positive regression dependence*).

Gilbert's modified Tarone-FDR procedure

Suppose there are m hypotheses $H_1, H_2, \ldots, H_m$, which result in m p-values $p_1, p_2, \ldots p_m$, which are all computed from contingency tables. Suppose also that we have pre-specified a significance threshold α.

The first step of Gilbert's modified FDR-Tarone Gilbert (2005) procedure is to apply Tarone's original procedure to the p-values (and the associated minimum attainable p-values α_i^*) to compute the integer K, the set R_K and the size of this set $m(K) = |R_K|$, i.e.

$$R_K = \{i \in \{1, 2, \ldots, m\} \,|\, \alpha_i^* < \alpha/K\}, \qquad m(K) = |R_K| \leq K.$$

The second step is to then perform the original FDR procedure Benjamini and Hochberg (1995a) on the subset of $m(K)$ hypotheses indexed in R_K. If necessary, relabel these hypotheses as $H_1, H_2, \ldots, H_{m(K)}$. Then, sort the p-values as in the original procedure:

$$p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m(K))},$$

and then find L, the largest i such that

$$p_{(i)} \leq \left(\frac{i}{m(K)}\right)\alpha.$$

Instead of α, one could use

$$\widetilde{\alpha} = \frac{\alpha}{\sum_{j=1}^{m(K)}(1/j)}, \tag{3}$$

as suggested in Benjamini and Yekutieli (2001), in order to account for special form of dependence.

Implementation of Tarone-FDR procedure with `FastCMH`

Gilbert's modified FDR-Tarone procedure can be easily integrated into the original `FastCMH` algorithm as detailed in Algorithm 1. Line 1 in Algorithm 1 finds δ_{tar} and $\mathcal{R}_T(\delta_{tar})$, using Tarone's procedure (here, δ_{tar} is actually α/K in Tarone's original notation). This step results in the set of testable hypotheses $\mathcal{R}_T(\delta_{tar})$ and their associated p-values.

Next, in the original algorithm Line 2 procedes with Tarone's method for determining which p-values are significant and which null hypotheses should be rejected.

However, in Algorithm 1* Line 2 is now the FDR step in Gilbert's modified procedure.

Line 3, as in the original Algorithm 1, is the filtering step that clusters overlapping genomic regions and selects the most significant region in a cluster, discarding the rest.

It is also possible to use the threshold in Equation (3) to create an algorithm `FastCMH-FDRDep`, which takes a form of dependence into account.

Experimental results comparing `FastCMH` with `FastCMH-FDR` on simulated data and the COPDGene study can be found in Sections S3.1.7 and S3.2.7, respectively.

Algorithm 1* `FastCMH-FDR`

Input: Dataset $\mathcal{G} = \{\mathbf{g}_i, y_i, c_i\}_{i=1}^{n}$, desired FWER α
Output: Set of non-overlapping conditionally associated genomic regions $\mathcal{R}_{sig,filt} = \{[\![t_s, t_e]\!] \,|\, p([\![t_s, t_e]\!]) \leq \delta_{tar}\}$
1: $(\delta_{tar}, \mathcal{R}_T(\delta_{tar})) \leftarrow$ `get_testable_regions`$(\mathcal{G}, \alpha)$
2: $\mathcal{R}_{sig,raw} \leftarrow$ FDR procedure on p-values indexed by $\mathcal{R}_T(\delta_{tar})$
3: $\mathcal{R}_{sig,filt} \leftarrow$ `filter_overlapping_regions`$(\mathcal{R}_{sig,raw})$
4: Return $\mathcal{R}_{sig,filt}$

S1.5 Burden tests

In this section, we describe the burden tests that were performed on the three types of data detailed in Section S2, i.e. simulated datasets, a case/control human dataset (COPDGene) and five plant datasets (*Arabidopsis thaliana*)

In general, two alternative encodings have been used to collapse the SNPs (or markers) in each candidate region into a single meta-marker:

(I) an indicator of the presence of any number of minor allele in the region, equivalent to the encoding used by `FastCMH`.

(II) the count of minor alleles in the region.

The different experimental designs of the burden tests, on each type of data in which they were performed, are described below.

S1.5.1 Simulated data

We performed two types of burden test on two different simulated datasets: window-based burden tests and gene-based burden tests. This section describes the two types of simulations and the burden tests ran on them. Section 4.1.2 in the main document presents the results of the simulations with window-based burden tests. The results on the simulated gene-based burden tests are described in Section S3.1.5.

Window-based burden tests: The datasets were generated as described in Section 4.1.2 of the main manuscript. We performed simulations using two types of windows, namely *non-overlapping* and *sliding* windows. For both of them, w was the size of the window that was tested and varied across the burden tests.

> For the burden tests with *sliding* windows, $inc = 1$ was the number of markers (or stride) between the starting positions of two consecutive windows. This is illustrated in Figure S3. In the literature (Schmid and Yang, 2008; Casale *et al.*, 2015), strides of length $inc \in [1, 20]$ are considered, as well as a stride of length $inc = w/2$, however the most common one is $inc = 1$. Additional simulations, which are not shown here, with different values of inc have been performed and lead to the same conclusion regarding the gain of power of `FastCMH` compared to the sliding window-based burden tests.
>
> For the burden tests with *non-overlapping* windows, the alignment of the window boundaries with the boundaries of the truly associated genomic regions has a strong influence on the power of the burden test. In the analysis described in Section S3.1.5, our goal is to compare `FastCMH` against the most favorable scenario for burden tests. Therefore, the starting position of all truly associated regions coincide with the starting position of one tested window. The windows can be smaller or bigger than the truly associated region, but because there will always be a window that starts in the same location as the region, this puts the burden tests in a more favorable position by minimizing the fragmentation of the genome in windows of size w.

To represent the biological diversity of causal regions each dataset included seven truly associated genomic regions of different length ℓ, with $\ell \in [2, 4, 6, 8, 10, 12, 14]$. The same layout was applied to the confounded genomic regions. When simulating the data, we ensured that the confounded regions were far apart from each other in order to have distinct signals for all associated regions. In all window-based burden tests, for a fixed p_s, we calculated the power, the false detection proportion of confounded regions and the FWER by averaging the results over all associated genomic regions and by correcting with a Bonferroni correction factor equal to the total number of tests performed (i.e. number of windows $\simeq l/inc$ where l is the total number of markers in the sequence). In this study we did not take into account the dependence between the tests. This is left for future work.

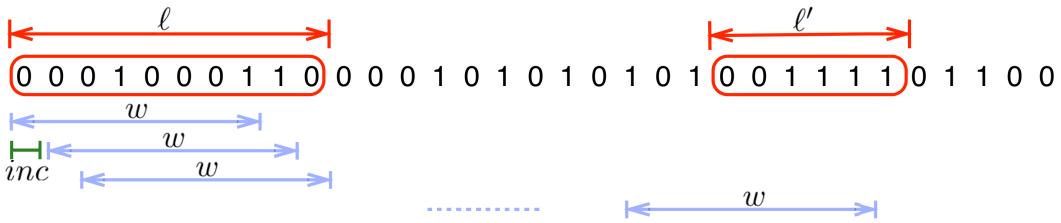

Figure S3: An illustration of the sliding window simulation with $w = 8$ genomic markers. The stride of one marker ($inc = 1$) between two consecutive windows is indicated in green. All of the windows are tested. Two truly associated genomic regions are represented in red with lengths $\ell = 10$ and $\ell' = 6$.

Gene-based burden tests: As opposed to window-based burden tests, gene-based burden tests do not take into account all genomic markers but only predefined regions of interest, normally based on prior biological knowledge. In the simulations, two associated regions are simulated, one truly associated with the phenotype and one confounded with the phenotype. In order to define the genomic regions that are tested – also referred to as *windows* in this analysis – two parameters w and f were used. The parameter w indicates the number of markers in each of the windows tested by the burden tests. One of the windows overlaps with the truly associated region and one window overlaps with the confounded region. The parameter f measures the overlap of a region (either the truly associated or the confounded) to a tested window. More precisely, f is equal to the proportion of the w markers of the tested window that are contained in a region (again, truly associated or confounded). The burden tests were performed under seven combinations of (w, f) as shown in Table S1. Figure S4 illustrates the interplay of the parameters w, ℓ and f in the simulated data.

Table S1: Combinations of parameters w and f used in simulation experiments for burden tests.

Case	Parameters	
	w	f
(a)	8	0
(b)	8	$1/4$
(c)	8	$1/2$
(d)	8	$3/4$
(e)	10	$4/5$
(f)	4	1
(g)	8	1

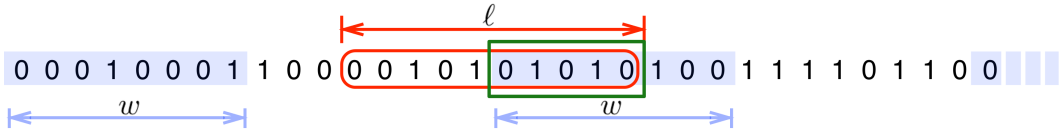

Figure S4: A simulation of a gene-based burden test. The burden tests will only be conducted on the w markers of the regions marked in blue. For simplicity, we assume that these regions correspond to genes and that all genes have the same number of markers ($w = 8$ in the figure). The region highlighted in red with length $\ell = 10$ is the truly associated genomic region. The overlap between the truly associated region and a gene is shown in green. The value of f is equal to the proportion of the w markers in the gene that are also contained in the truly associated genomic region. In this example, $f = \frac{5}{8}$.

S1.5.2 Human data: COPDGene

We conducted two sets of experiments using burden tests on human data. For both we used the encodings (I) and (II) described above to combine the markers in the genomic regions of interest, however these two sets of experiments differed by the type of genomic regions that were analyzed.

(a) As candidate genomic regions we considered all genes that overlap with at least one marker[1] This resulted in 17,817 regions for COPDGene.

(b) As candidate genomic regions, we partitioned the genome into contiguous non-overlapping windows of: i) 500 kilobases and ii) 1 megabase. Burden tests were performed on the SNPs that overlapped with these windows

[1] A marker is considered to overlap with a gene if it is located within 10 kb of the gene boundaries.

using the same encodings (I) and (II) mentioned above. It is worth noting that in this setup, all SNPs are being tested by being included in one and only one genomic window.

All types of burden tests were performed using both the likelihood ratio test under a logistic regression model, with the categorical covariate encoded using k dummy indicator variables, as well as using the CMH test in the same way `FastCMH` does[2]

S1.5.3 Plant data: *Arabidopsis thaliana*

As it was the case for the burden tests on human data, here we also used the encodings (I) and (II) to combine the markers in the genomic regions of interest. As candidate genomic regions, we considered:

(a) All genes that overlap with at least one marker (same mapping as in human data). This resulted in 24,426 regions for *Arabidopsis thaliana*.

(b) Partitions of the genome into contiguous non-overlapping windows of: i) 500 kilobases and ii) 1 megabase. Burden tests were performed on the SNPs that overlapped with these windows.

The burden tests were performed using both the likelihood ratio test under a logistic regression model, with the categorical covariate encoded using k dummy indicator variables, as well as using the CMH test in the same way `FastCMH` does. In the case of the *A. thaliana* datasets, we performed the previously mentioned burden tests as well as a last one that uses the three largest principal components from the kinship matrix as covariates (Price *et al.*, 2006) for both encodings (I) and (II).

[2]The CMH test only applies to encoding (I), which is binary.

S2 Datasets

This section provides additional information of the datasets used throughout the experiments. In Section S2.1, details about the data generation process are provided for all simulation experiments. Additional characteristics of the human and plant data are also presented in Sections S2.2 and S2.3. Finally, Section S2.4 discusses how the categorical covariates were defined for both human and plant data.

S2.1 Simulated data

Here, we describe in detail the model used to generate synthetic data for each of the simulations included in this article. Firstly, in Section S2.1.1, we present the basic simulation model employed for the experiments of Section 4.1 in the main manuscript. Next, Section S2.1.2 describes how we extended the previous model to include a simple form of linkage disequilibrium, leading to the results presented in Section S3.1.4.

S2.1.1 Basic simulation model: generating truly significant and confounded genomic regions

In the simulation study in Section 4.1, Figures 2(a) and 2(b) were created using a dataset where each sample had one truly significant genomic region and one confounded genomic region. This section provides further details on how these sequences and this dataset were constructed.

In order to investigate the difference in power between the `FastCMH`, `FAIS-`χ^2 and `BonfCMH`, we construct a simulated dataset consisting of n individuals with l binary markers each. Besides, we consider a covariate with $k = 2$ categories for each sample.

Initially each element $g_i[t]$ is sampled i.i.d. from a Bernoulli distribution with probability p_1 of being a 1, i.e. $g_i[t] \sim \mathrm{B}(1, p_1)$. In our case, we set $p_1 = 0.3$. One could consider this to be the "background noise". Next, in each sample, two significantly associated genomic regions will be included: $[\![t_{s,sig}, t_{e,sig}]\!]$ and $[\![t_{s,con}, t_{e,con}]\!]$. Note that in our experiment, the length of each region, denoted by ℓ, is the same. The first one, $[\![t_{s,sig}, t_{e,sig}]\!]$, will be directly associated with the phenotype y. However, the other one, $[\![t_{s,con}, t_{e,con}]\!]$, will be associated with y merely via the covariate c (see Figure S5). To generate data according to that dependence structure, we initially sample (R package `bindata`) three random binary variables $intSig$, c and y according to a (three-dimensional) multivariate Binomial distribution with mean vector μ and covariance matrix Σ,

$$\mu = (0.5, 0.5, 0.5), \qquad \Sigma = \begin{bmatrix} 1 & 0 & p_{\mathrm{s}}/2 \\ 0 & 1 & \rho_{\mathrm{con}}/2 \\ p_{\mathrm{s}}/2 & \rho_{\mathrm{con}}/2 & 1 \end{bmatrix}.$$

The rows/columns of Σ correspond to:

- the first row/column represents the genomic marker $g([\![t_{s,sig}, t_{e,sig}]\!])$ of the truly significant genomic region.
- the second row/column represents the value of the categorical covariate c (which has two classes, i.e $k = 2$).
- the third row/column represents the case/control status, y.

To be clear, for each sample $i \in \{1, 2, \ldots, n\}$,

$$(intSig_i, c_i, y_i) \sim \mathrm{B}(1; \mu, \Sigma).$$

Since the mean vector μ has all its components equal to 0.5, there will be approximately $n/2$ samples with $y = 0$ (controls) and $n/2$ samples with $y = 1$ (cases). Also, approximately $n/2$ of the values for $intSig$ will be 1, and $n/2$ will be zero. Let us consider the vectors $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, $\mathbf{intSig} = (intSig_1, intSig_2, \ldots, intSig_n)$, $\mathbf{c} = (c_1, c_2, \ldots, c_n)$. By our choice of Σ, if the parameter p_{s} is close to 1, then the vectors $\mathbf{y}$ and $\mathbf{intSig}$ will be highly correlated, i.e. when $y_i = 1$, then $intSig_i$ will be 1 with high probability, and when $y_i = 0$, then $intSig_i$ will be 0 with high probability.

Now, a random location for the interval $[\![t_{s,sig}, t_{e,sig}]\!]$ is chosen, with length ℓ. If $intSig_i = 1$, all values in $[\![t_{s,sig}, t_{e,sig}]\!]$ are set to zero (recall that some of these binary values may be equal to 1, having been generated by $\mathrm{B}(1, p_1)$), except one randomly chosen index j, $t_{s,sig} \leq j \leq t_{e,sig}$, which is set to be 1. So, a truly significant interval (when $intSig_i = 1$) contains exactly one 1. On the other hand, when $intSig_i = 0$, then all binary values in the randomly chosen interval will be zero. This explains the construction of the truly significant genomic regions.

For the confounded intervals, the genomic marker of the confounded genomic region $g([\![t_{s,con}, t_{e,con}]\!])$, denoted $intCon$, is obtained from the values of the categorical covariate c by flipping its value with a low probability $p_\epsilon = 0.05$. To be clear, we sample $\epsilon \sim \mathrm{B}(1, p_\epsilon)$ and then

$$intCon = c \oplus \epsilon,$$

where $\oplus$ is the **xor** operator. By looking at Σ, we see that the parameter ρ_{con} controls the degree of association between c and y. For high values of ρ_{con}, the vectors $\mathbf{c}$ and $\mathbf{y}$ will be highly correlated.

Similarly to the case for the truly significant genomic region, when $intCon = 1$, an interval $[\![t_{s,con}, t_{e,con}]\!]$ containing exactly one 1 is inserted into the sequence of genomic markers (and when $intCon = 0$, all the values in the genomic region are zero). Note that the significant and confounded regions are placed at random locations in the sequence of genomic markers, with the only constraint being that they do not overlap.

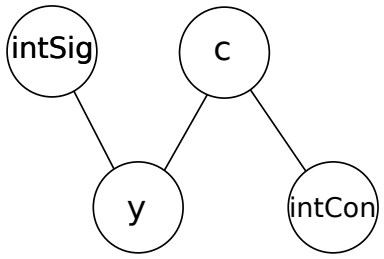

Figure S5: A graphical model indicating the dependencies between significant regions, the labels, the confounding variable and the confounded region. Here, *intSig* is an indicator variable indicating whether or not a given sample has at least one minor allele in a truly significant genomic region, y is the case/control phenotype for the sample, c is the category of the confounding variable for the sample, and *intCon* is an indicator variable indicating whether or not the sample contains at least one minor allele in the confounded genomic region.

Creating Figures 2(a) and 2(b) in the main manuscript

For a single experiment, a dataset is generated with parameters: $n = 500$, $l = 1,000,000$, $p_1 = 0.3$, $k = 2$ and the length of the significant/confounded regions is set to be $\ell = 5$. Furthermore, $\rho_{\text{con}} = p_{\text{s}}$ as p_{s} varies in the interval $[0.1, 0.9]$. Running the algorithms over this data set provide us with a set of values. This experiment is repeated 500 times to get average values which are then plotted in Figures 2(a) and 2(b).

Creating Figure 2(c) in the main manuscript

The parameters used to create Figure 2(c) were: $n = 500$, $p_1 = 0.3$, $k = 2$ and l varies between $l = 100$ and $l = 10,000,000$. Note that no significant/confounded regions were generated for this runtime plot (it was found in earlier experiments that having/not having significant and confounded regions made no difference to the runtime results).

Creating Figure 2(d) in the main manuscript

The parameter values used to create Figure 2(d) were: $n = 500$, $l = 100,000$, $p_1 = 0.3$ and k varies between $k = 1$ and $k = 30$. Note that no significant/confounded regions were generated for this runtime plot (it was found in earlier experiments that having/not having significant and confounded regions made no difference to the runtime results).

S2.1.2 Incorporating linkage disequilibrium into the model

We created an additional simulation dataset that incorporates a simplified model of linkage disequilibrium by splitting the sequence of l markers into b disjoint blocks with $l_b = l/b$ markers each. Markers within each block have pairwise correlation ρ_{ld}, whereas distinct blocks are uncorrelated. This results in genotype sequences that exhibit a blockwise correlation structure, with the simulation parameter ρ_{ld} controlling the strength of linkage disequilibrium inside each block. The minor allele frequency of each marker was independently sampled from a uniform distribution in the range $[0.01, 0.20]$.

Among the b disjoint blocks, two blocks are chosen to contain the truly associated and confounded signals, respectively. The markers in the center of each block were used to generate the covariate and case/control status. More precisely, the covariate is obtained by adding a small corruption to the central marker of the "confounded" block, in the same manner as discussed in Section S2.1.1. Then, the case/control status is generated such that its correlation with the covariate is $\rho_{con}/2$, and the correlation with the central marker of the "truly associated" block is $\rho_s/2$. Most importantly, in this simulation, both central markers are removed from the dataset. This implies that both the truly associated signal and the confounded signal are only present in the observed markers indirectly via linkage disequilibrium.

Using the model described in the paragraph above, we generated synthetic data corresponding to $n = 500$ individuals genotyped at $l = 1,000,000$ markers, split into $b = 10,000$ disjoint blocks of $l_b = 100$ markers each. The signal strength was varied in the range $\rho_s = \rho_{con} \in [0.05, 0.95]$. To illustrate the effect of changing the linkage disequilibrium strength, four different values for ρ_{ld} were used: $\rho_{ld} \in \{0.175, 0.25, 0.375, 0.5\}$.

S2.2 Human data: COPDGene

The COPDGene dataset contains samples from two populations: African Americans and non-Hispanic whites. For each population, samples with missing phenotypes (i.e. not labeled as either case or control) or with missing height information were removed from the analysis. Additionally, SNPs were filtered out if: i) minor allele frequency $<$ 0.01, or ii) Hardy-Weinberg equilibrium $<$ 1.0e-6. Missing genotypes were imputed as described in (Cho *et al.*, 2014). SNPs were binarized using a dominant encoding: homozygous major SNPs were encoded as 0 while heterozygous and homozygous minor SNPs were both encoded as 1. Table S2 shows the number of samples in each of the cohorts.

Table S2: Details about the samples in each population of COPDGene. The column "case" refers to individuals who were diagnosed with COPD. The number of SNPs in the intersection of both populations is 615,906 and these were the SNPs used in our analysis.

Population	Disease status			Gender	
	case	control	Total	male	female
African Americans	821	1,826	2,647	1,498	1,149
non-Hispanic whites	2,812	2,534	5,346	2,816	2,530
Total	3,633	4,360	**7,993**	4,314	3,679

We performed a principal component analysis (PCA) to highlight the genetic differences between the individuals in the African American and non-Hispanic white groups. Using the first two principal components, we plotted all samples and Figure S6 shows their clustering.

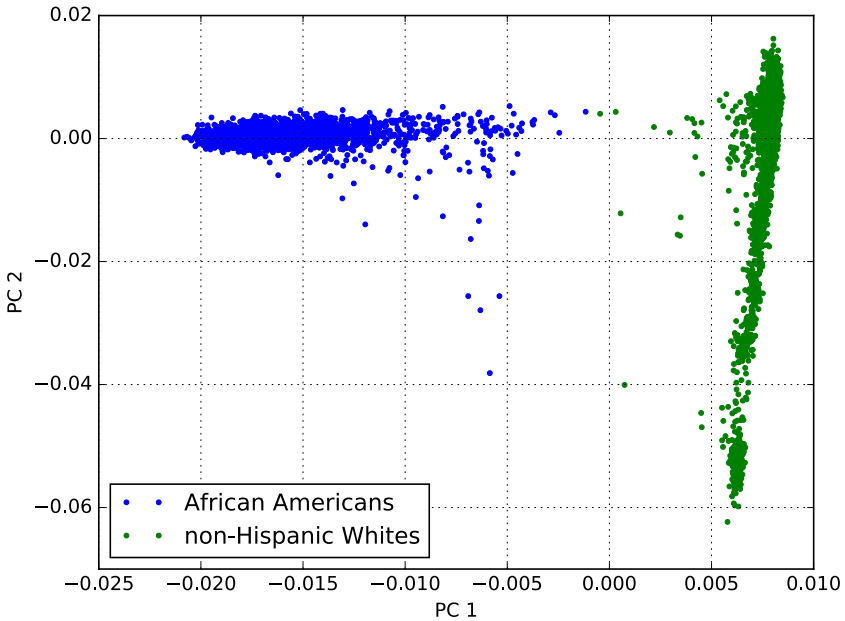

Figure S6: Embedding of the 7,993 samples in COPDGene according to the first two principal components.

S2.3 Plant data: *Arabidopsis thaliana*

In order to provide a meaningful comparison with state-of-the-art algorithms, we chose five phenotypes for which at least two significant genomic regions were identified in Llinares-López *et al.* (2015) –without any correction for population structure. Additionally, the phenotypes should contain a relatively balanced number of cases and controls (ratio between $80\% - 20\%$). For this study, we did not filter SNPs by minor allele frequency. Homozygous minor SNPs were encoded as 1 whereas homozygous major SNPs were encoded as 0. No heterozygous SNPs were present in the data. Table S3 provides information about the number of SNPs and samples for each phenotype.

Table S3: Samples for which the five different phenotypes we analyzed was known. Depending on the phenotype of interest, the number of SNPs varied. For each phenotype, the number of samples labeled as "case" vs. the controls fluctuated between 22% (LES) and 66% (*avrRpm1*).

Phenotype	Number of SNPs	Sample label		
		case	control	Total
avrB	214,032	55	32	87
avrRpm1	214,022	56	28	84
avrPphB	214,032	46	44	90
LES	214,051	21	74	95
LY	214,051	29	66	95

S2.4 Definition of covariates for COPDGene and *Arabidopsis thaliana*

In a GWAS, spurious associations between genotype and the trait of interest can be found due to confounding factors such as gender, age or other phenotypic characteristics. A particularly common source of confounding in GWASs is population structure (Marchini *et al.*, 2004), a phenomenon that occurs when the dataset contains individuals with genetic ancestry from distinct subpopulations exhibiting systematic genetic differences (e.g. different allele frequencies). When the trait of interest (e.g. disease prevalence) happens to also differ among subpopulations, population structure leads to many spurious associations. The ability of `FastCMH` to handle categorical covariates can be used to correct for confounding due to population structure. While defining the covariate is straightforward in datasets for which the race or ethnicity of the individuals is known, as in the COPDGene study, `FastCMH` can also be successfully applied in cases in which population structure acts as a hidden variable, as occurs in all five *A. thaliana* datasets.

COPDGene

In the COPDGene study, the categorical covariate c can be based on the (known) genetic ancestry of the individuals, namely African Americans and non-Hispanic whites. To illustrate both the ability of `FastCMH` to cope with several covariates simultaneously and to handle a large number of categories k for each covariate, we also consider "height". Height has been used as a covariate in previous COPD studies to serve as a proxy for lung volume. To define the categorical covariate c, the range of observed values for height across the entire dataset is split into ten deciles and we set $c_i = h, h \in \{1, \ldots, 10\}$ for African American individuals with height corresponding to the h-th decile, and $c_i = 10 + h, h \in \{1, \ldots, 10\}$ for non-Hispanic white individuals with height corresponding to the h-th decile.

A. thaliana

For each of the five *A. thaliana* datasets, the categorical covariate c we condition on to correct for population structure was defined using k-means clustering on the three principal components (PCs) of the empirical kinship matrix (Price *et al.*, 2006). We selected the best value of k in a range from 2 to 5 according to the resulting genomic inflation factor independently for each dataset. In this way, for each sample $i \in \{1, \ldots, n\}$, we define $c_i = h$ if k-means assigned the sample to the h-th cluster.

S3 Supplementary Results

This section presents additional results for each of the three types of data that were used in our analysis: 1) simulated data, 2) COPDGene and 3) *Arabidopsis thaliana*.

S3.1 Simulation study results

Here we present results for a number of additional simulation experiments complementary to those described in Section 4.1 of the main manuscript.

S3.1.1 Type I error control

Due to the use of Tarone's method, `FastCMH` is theoretically guaranteed to control the Family-Wise Error Rate (FWER). In this section, we illustrate this property with empirical results. Figure S7 shows the FWER corresponding to the experiments described in Section 4.1.1 of the main manuscript. Those were performed on synthetic data generated as described in Section S2.1.1 with $n = 500$ samples, $l = 1,000,000$ markers, $k = 2$ categories for the covariate and $p_1 = 0.3$. The signal strength, which was set to be identical for the truly associated and confounded genomic regions, was varied in the range $\rho_s = \rho_{con} \in [0.05, 0.95]$. The target FWER was set to $\alpha = 0.05$ following standard practice. The results here shown were obtained after averaging 500 repetitions of the experiment.

As shown in the figure, the actual FWER for `FastCMH`, `FAIS-`χ^2 and `BonfCMH` is indeed below this threshold, empirically confirming FWER-control. `BonfCMH`, which does not employ Tarone's method, can be seen to be greatly over-conservative, leading to a significant loss of statistical power as shown in Figure 2(a) in the main manuscript. In contrast, by using Tarone's method, `FastCMH` and `FAIS-`χ^2 are only slightly over-conservative, being able to retain a large statistical power. `FastCMH` and `FAIS-`χ^2 could be made even less over-conservative by employing a criterion that takes into account the dependence existing between test statistics, such as permutation-testing based approaches.

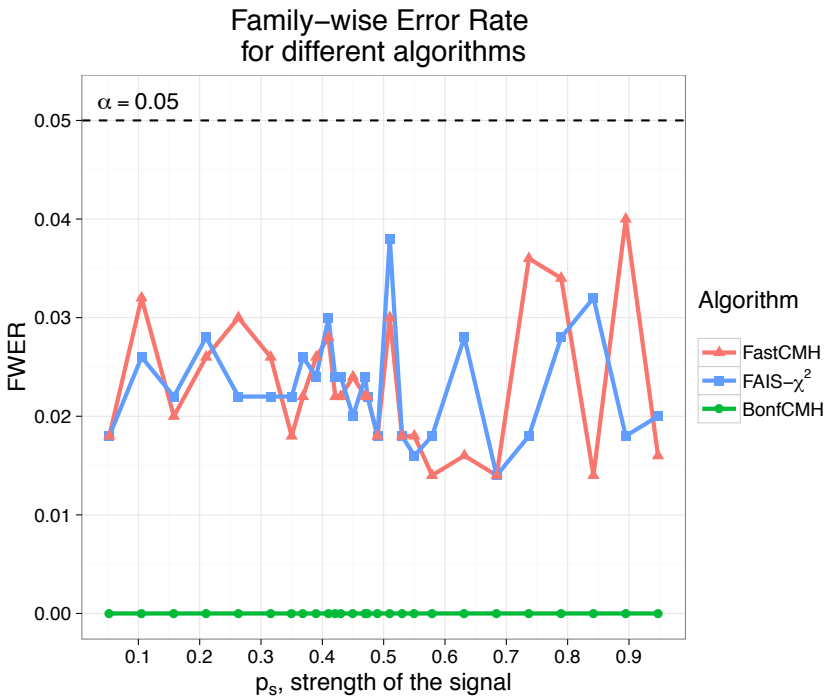

Figure S7: The actual Family-wise Error Rate of `FastCMH`, `FAIS-`χ^2 and `BonfCMH`, when each algorithm specifies the target FWER to be $\alpha = 0.05$.

Finally, Figure S8 illustrates Type I error control by comparing the QQ-plots for `FastCMH` and `FAIS-`χ^2 in two representative scenarios: (a) the global null model, corresponding to $\rho_s = \rho_{con} = 0$ and (b) a model with a single confounded genomic region with signal strength $\rho_{con} = 0.8$ but no truly associated genomic region, i.e $\rho_s = 0$. As expected, the distribution of p-values obtained with `FastCMH` agrees with the expected null distribution in both scenarios (a) and (b). In contrast, the p-value distribution obtained with `FAIS-`χ^2 shows a mild inflation in scenario (b), due to existence of one confounded genomic region. This deviation from the expected null would have

been more severe if more confounded regions had been included in the simulation, since FAIS-χ^2 cannot correct for covariates as FastCMH does.

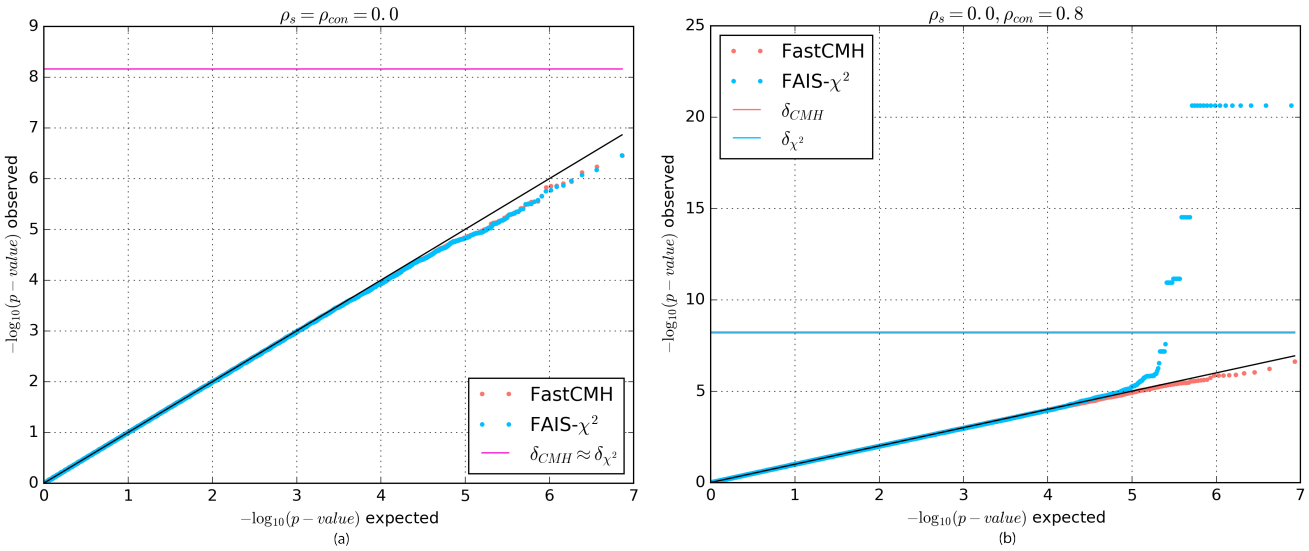

Figure S8: Comparison of the QQ-plots for the p-values of all testable genomic regions in simulated data. (a) global null model, $\rho_s = \rho_{con} = 0$. (b) a single confounded genomic region and no truly associated genomic region, $\rho_s = 0, \rho_{con} = 0.8$.

S3.1.2 Power, false detection proportion and FWER as a function of the number of categories

In this section, we explore the effect that the number of categories for the covariate, k, has on the power, false detection proportion and FWER of FastCMH and its comparison partners. To this end, we first generated data according to the model described in Section S2.1.1, using $n = 500$ individuals, $l = 1,000,000$ markers and $p_1 = 0.3$. The strength of the signal and confounding were fixed to $\rho_s = \rho_{con} = 0.5$. In these experiments, the categorical covariate was defined for each individual by assigning them a random category in the set $c_i \in 1, 2, \ldots, \frac{k}{2}$ if the (binary) covariate generated as described in Section S2.1.1 takes value 1 for the individual, and a random category in the set $c_i \in \frac{k}{2} + 1, \ldots, k$ if it takes value 0. The value of the number of categories for the covariate, k, was varied in the range $k \in [\![2, 18]\!]$. All results were obtained by averaging 500 repetitions of the experiment.

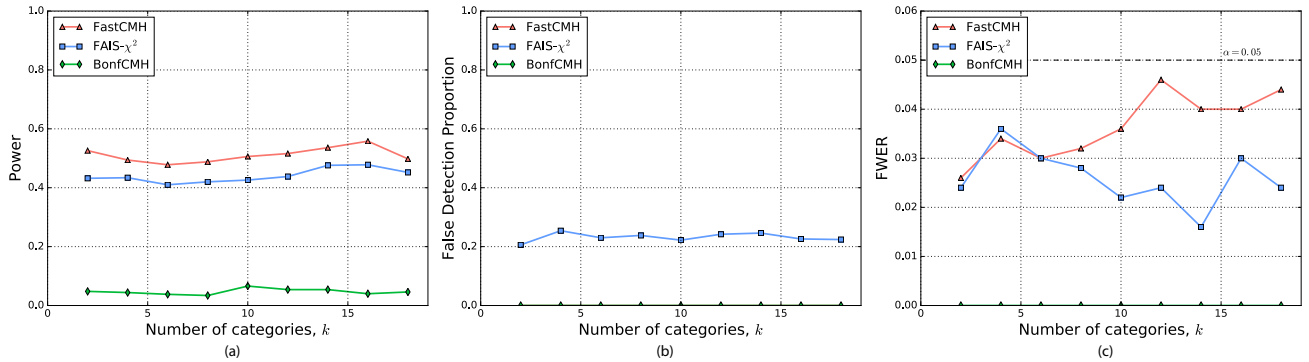

Figure S9: A comparison of (a) statistical power, (b) proportion of confounded regions falsely detected and (c) FWER for FastCMH, FAIS-χ^2 and BonfCMH as the number of categories for the covariate, k, varies.

As Figure S9(a) shows, the power of both FastCMH and FAIS is approximately independent of the number of categories, provided that the average number of observations per contingency table is not too small (e.g. ≥ 15). In Figure S9(b) we can observe that the effectiveness of FastCMH to correct for confounders is unaffected by k, while Figure S9(c) shows that strict FWER control is retained for all k.

S3.1.3 Scaling of the runtime with respect to the number of samples in the dataset

In Section 4.1.1 of the main manuscript, we showed that the runtime of `FastCMH` scales gently with respect to the number of markers, l, and the number of categories of the covariate, k. In this section, we complement those results by studying how `FastCMH` scales in terms of runtime with respect to the number of samples in the dataset, n. Figure S10 shows that the computation time of `FastCMH`, `FAIS-`χ^2 and `BonfCMH` increases approximately linearly with the number of samples, n. Consistent with the results shown in the main manuscript, `FastCMH` and `FAIS-`χ^2 show a similar performance in terms of runtime as the sample size varies. However, the absolute runtime is much larger for `BonfCMH`, which does not employ Tarone's method to efficiently prune the search space. This becomes clearer in Figure S11, which depicts the runtime of `FastCMH` and its two comparison partners in log-scale. For instance, `FastCMH` and `FAIS-`χ^2 take approximately two minutes to process a data set with $n = 10,000$ samples, while `BonfCMH` takes over 24 hours to process the same data set. In all figures, we used $k = 4$, $l = 100,000$ and $p_1 = 0.3$.

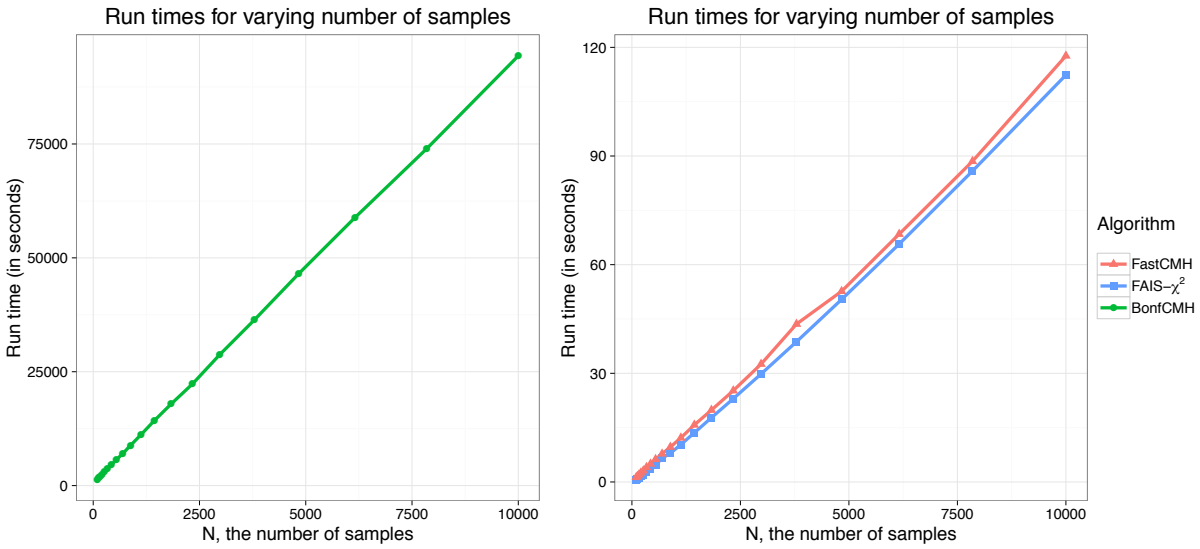

Figure S10: The run times of `FastCMH`, `FAIS-`χ^2 and `BonfCMH`, as the number of samples, n, varies.

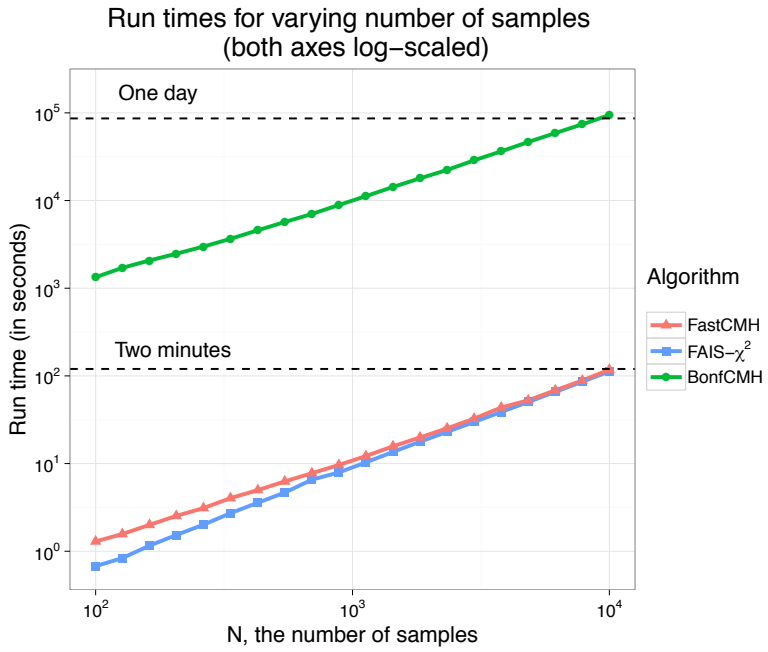

Figure S11: The run times of `FastCMH`, `FAIS-`χ^2 and `BonfCMH`, as the number of samples, n, varies. In this figure, both axes are log-scaled.

S3.1.4 Taking linkage disequilibrium into account

In this section we extend our results on the simulated data that incorporates a form of linkage disequilibrium as described in Section S2.1.2. In particular, the main goal is to assess the performance of `FastCMH` in a scenario where the causal variants are not included among the observed markers. To that effect, we compare our method against `FAIS-`χ^2, `BonfCMH` and single-SNP testing.

Unlike under the data-generation model used in the main manuscript, here the different markers in the associated genomic regions do *n*ot contain independent (weak) signals. Instead, the markers can be understood as tagging variants for the hidden causal variant of the region, where the amount of signal preserved depends on ρ_{ld}. In this setting, single-SNP testing can be expected to achieve good performance, specially if ρ_{ld} is sufficiently large. In principle, `FAIS-`χ^2 and `FastCMH` do not benefit from aggregating the markers in the region as much as in the previous model, since all variants being aggregated carry essentially the same signal. Nonetheless, the resulting meta-marker might still exhibit a stronger association than any of the individual variants in the region, as it acts as an "smoothed" tagging variant.

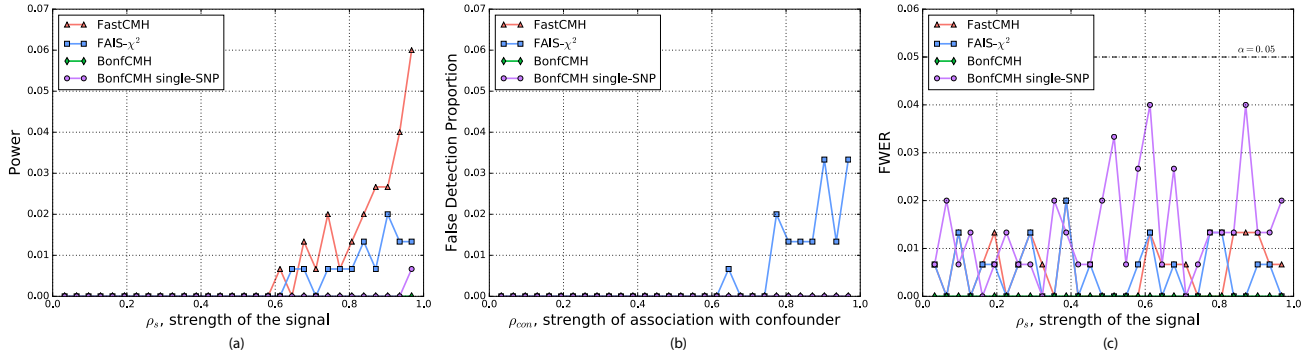

Figure S12: A comparison of (a) statistical power, (b) proportion of confounded regions falsely detected and (c) FWER for `FastCMH`, `FAIS-`χ^2, `BonfCMH` and single-marker tests (`BonfCMH`, single-SNP) with $\rho_{ld} = 0.175$.

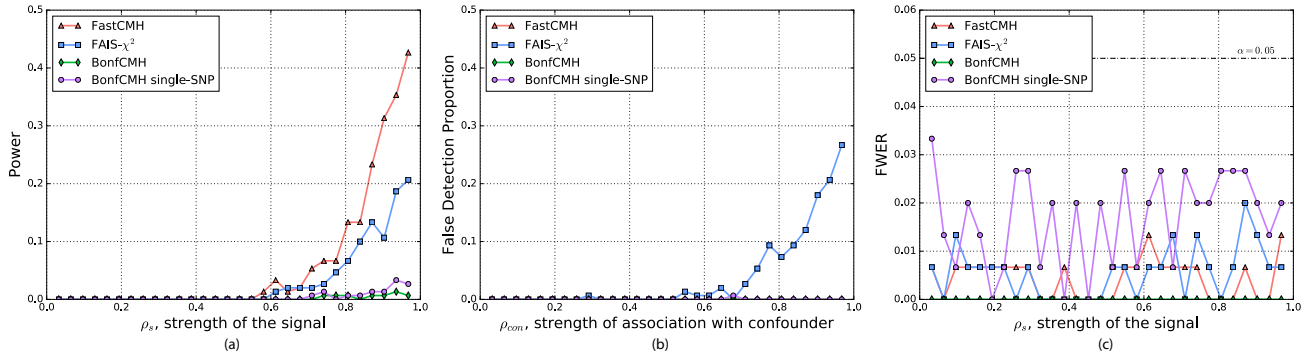

Figure S13: A comparison of (a) statistical power, (b) proportion of confounded regions falsely detected and (c) FWER for `FastCMH`, `FAIS-`χ^2, `BonfCMH` and single-marker tests (`BonfCMH`, single-SNP) with $\rho_{ld} = 0.25$.

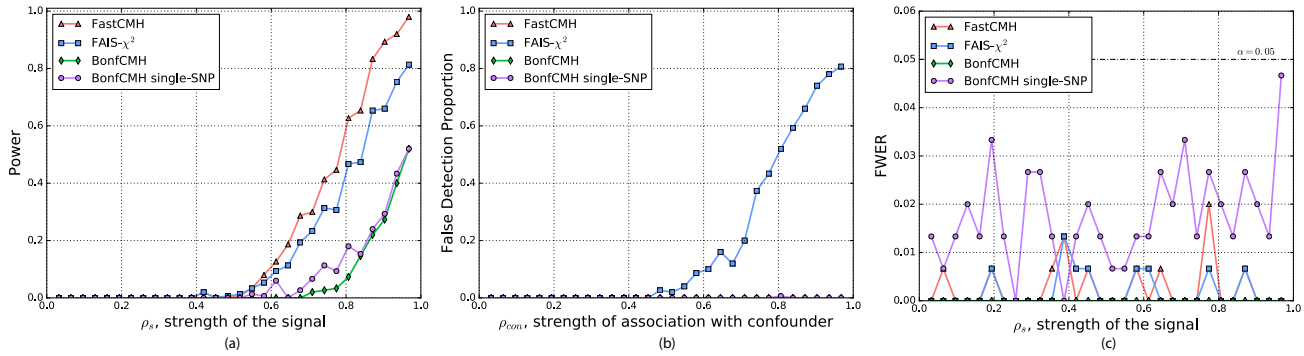

Figure S14: A comparison of (a) statistical power, (b) proportion of confounded regions falsely detected and (c) FWER for `FastCMH`, `FAIS-`χ^2, `BonfCMH` and single-marker tests (`BonfCMH`, single-SNP) with $\rho_{ld} = 0.375$.

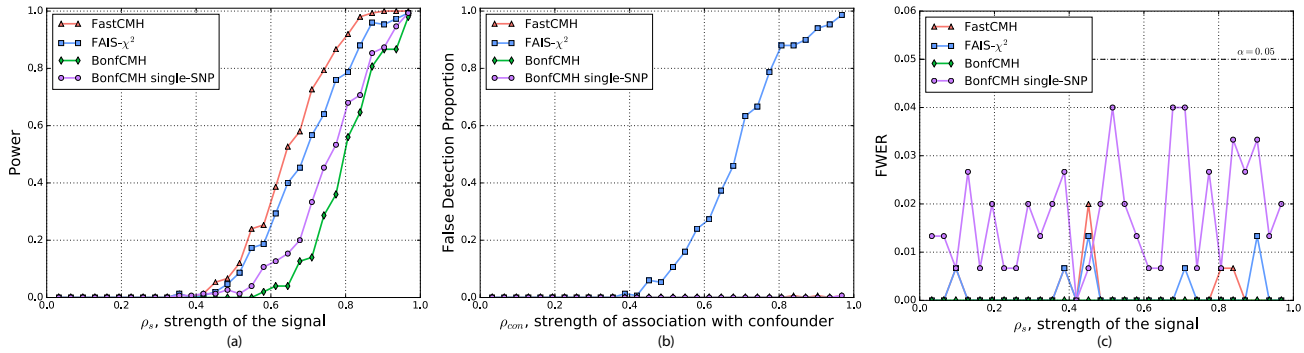

Figure S15: A comparison of (a) statistical power, (b) proportion of confounded regions falsely detected and (c) FWER for `FastCMH`, `FAIS-`χ^2, `BonfCMH` and single-marker tests (`BonfCMH`, single-SNP) with $\rho_{ld} = 0.5$.

We present our results based on the parameters used to generate the data described in Section S2.1.2, i.e. $n = 500$ individuals with $l = 1,000,000$ markers, split into $b = 10,000$ disjoint blocks of $l_b = 100$ markers each. Four different values for ρ_{ld} were used: $\rho_{ld} \in \{0.175, 0.25, 0.375, 0.5\}$. We averaged 150 repetitions of the experiment to obtain the results shown here. Figures S12-S15(a) confirm the hypothesis stated in the previous paragraph. While single-marker testing has non-zero power, `FastCMH` and `FAIS-`χ^2 are able to to exploit the existing linkage disequilibrium by aggregating multiple tagging variants, resulting in a noticeable improvement in statistical power. As expected, the effect is particularly pronounced for moderate linkage, e.g. $\rho_{ld} \in [0.2, 0.4]$.

Figures S12-S15(b), which show the probability of retrieving a confounded interval, illustrate that `FastCMH`'s ability to correct for confounding is unaffected by the presence of linkage disequilibrium. This is in contrast with `FAIS-`χ^2, which continues to be unable to deal with confounding.

Finally, Figures S12-S15(c) confirm that FWER control is not lost due to the presence of linkage disequilibrium. This is unsurprising, since Tarone's method guarantees FWER control regardless of the underlying dependence structure of the data.

S3.1.5 Burden tests

Here we present additional results to the ones presented in Section 4.1.2 of the manuscript. We first describe the results of the simulations that compare `FastCMH` with the burden tests that use non-overlapping windows. Secondly, we discuss the results of the burden tests with sliding windows. In both cases, markers in a window are collapsed using the encoding Enc. (I) presented in Section S1.5. Results based on Enc. (II) are not presented here because they are very similar to the results obtained using Enc. (I). We then compare our method `FastCMH` to simulated gene-based burden tests that use both encodings Enc. (I) and Enc. (II). For all methods, the target FWER was set to $\alpha = 0.05$. All experiments were performed with $n = 500$ samples and $l = 100,000$ markers. All tests were averaged over 200 iterations.

False detection proportion and FWER in window-based burden tests: Figures S16 (non-overlapping windows) and S17 (sliding windows) show in the y-axis an evaluation of two different metrics for the burden tests:

(a) the proportion of confounded regions falsely discovered and (b) the FWER. The x-axis represents the strength of the association p_s between the associated regions – true and confounded – and the phenotype. In Figures S16(a) and S17(a), all conditional tests succeed in not discovering the confounded genomic regions while FAIS-χ^2, which does not condition on confounders, reports them. Figures S16(b) and S17(b) show that almost all tests ensure a good control of the FWER; in particular FastCMH has a FWER below the threshold $\alpha = 0.05$.

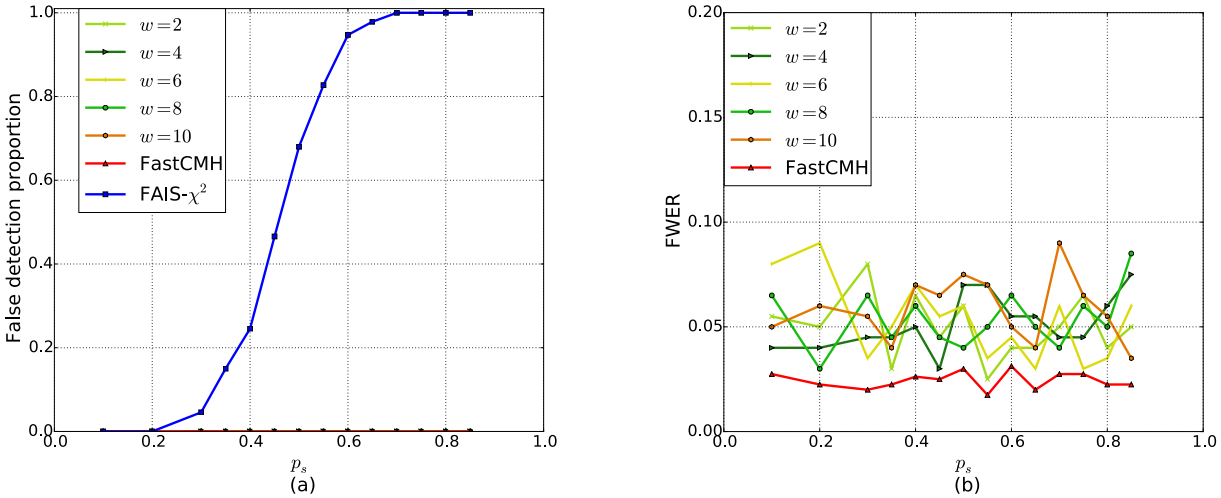

Figure S16: A comparison between burden tests with non-overlapping windows, FAIS-χ^2 and FastCMH: (a) proportion of confounded regions falsely detected (false detection proportion) and of (b) the FWER for FastCMH and for the burden tests. The value of w is the length of the windows in the burden tests and varies between 2 and 10.

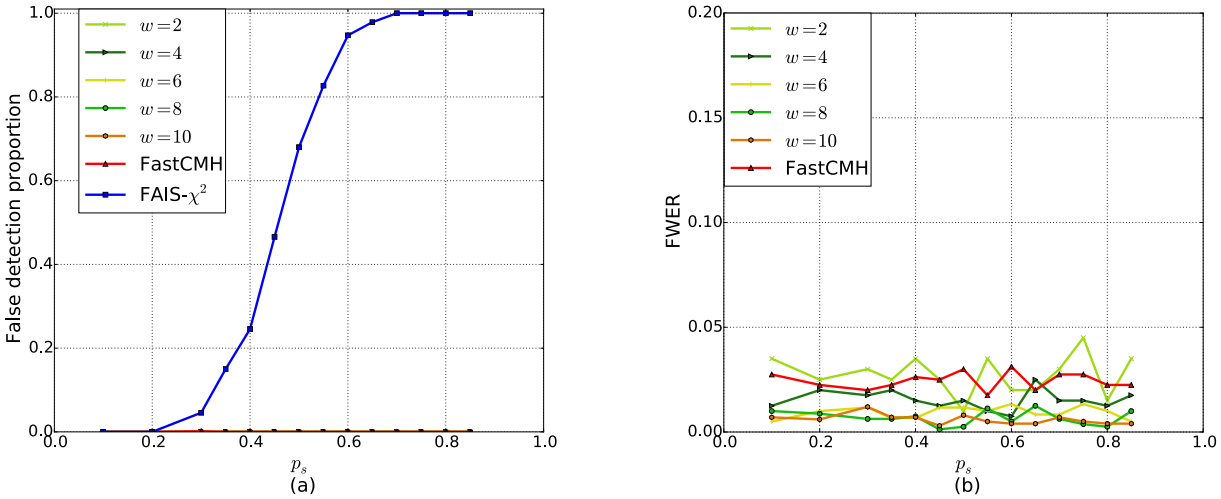

Figure S17: A comparison between burden tests with sliding windows, FAIS-χ^2 and FastCMH: (a) proportion of confounded regions falsely detected (false detection proportion) and of (b) the FWER for FastCMH and for the burden tests. The value of w is the length of the windows in the burden tests and varies between 2 and 10.

Influence of the length of the associated genomic regions on the results of window-based burden tests: We performed an additional set of experiments to evaluate the impact of the length ℓ of the associated genomic regions on the statistical power of the burden tests with non-overlapping and sliding windows.

Figure S18 shows how ℓ strongly impacts the power of the burden tests with non-overlapping windows of size w. We observe that, except in some rare configurations of parameters ($w = \ell$), FastCMH outperforms the burden tests in all settings, independently of the lengths of the window and of the associated region. In the simulations, the case $w = \ell$ is very beneficial to the burden tests using non-overlapping windows, in particular in the setup we chose where each associated genomic region is perfectly aligned with one tested window. In this setting, the non-overlapping windows achieve a similar power to that of FastCMH. However, FastCMH has better power in all other situations, for example when the length of the associated region does not match exactly the length of the tested windows (Figure S18) or if the tested window is not perfectly aligned with the associated region. In particular,

the statistical power of the burden tests drop when $|\ell - w|$ increases. In practice, neither the location of the truly associated genomic regions nor their length are known a priori. Thus, different tests with non-overlapping windows of different length have to be performed, leading to a loss of power as the Bonferroni correction becomes larger.

Figure S19 shows how the length ℓ of the associated genomic regions has an impact in the power of the burden tests with sliding windows of size w. `FastCMH` clearly outperforms the burden tests in all settings, independently of w and ℓ. For a fixed window-size w, the statistical power of the burden tests vary with the length of the associated region ℓ. Indeed, for each length of the associated region, it partially or fully overlaps with several windows. The distribution of the partial or full overlaps of the tested windows with respect to the associated region, depends on: a) the stride (inc) between two consecutive windows, b) the length of the associated region ℓ and c) the size of the windows w. These three factors strongly influence the power of the window-based tests. For example, if the window is large compared to the size of the associated region $w > \ell$ (cases $w = 6$ with $\ell = 2$ and $\ell = 4$), the power of the tests has a dramatic drop as the window is composed of irrelevant markers that contaminate the signal with noise. If $w \ll \ell$ (cases $w = 2$ with $\ell = 10$ and $\ell = 12$ and case $w = 4$ with $\ell = 12$), the windows do not contain enough of the truly associated markers to be significantly associated with the phenotype and the burden tests also perform poorly in these cases. The power of the burden tests with sliding windows increases when the overlapping windows are both small enough to only include associated markers and large enough to include a large fraction of the signal, so that the region can be detected (case $w = 2$ with $\ell = 4$ and case $w = 4$ with $\ell = 6$).

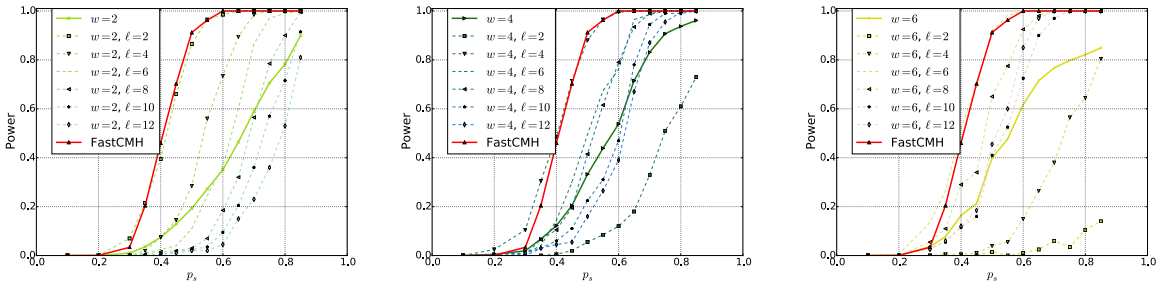

Figure S18: Comparisons of statistical power for `FastCMH` and burden tests with *non-overlapping* windows, for different lengths w of the windows and of the truly associated genomic region ℓ. The thick curve indicates the average of the statistical power across all burden tests for all genomic regions lengths $\ell \in [2, 4, 6, 8, 10, 12, 14]$. The thin dashed curves represent the power of the burden tests for each length of the associated region separately.

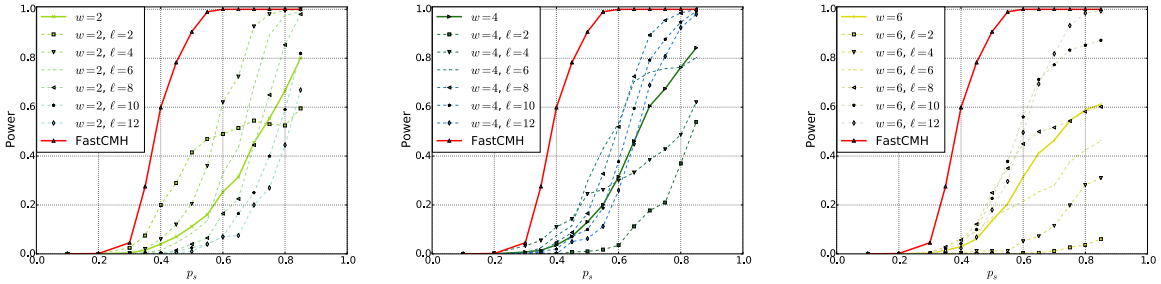

Figure S19: Comparisons of statistical power for `FastCMH` and burden tests with *sliding* windows, for different lengths w of the windows and of the truly associated genomic region ℓ. Two starting positions of two neighboring windows are separated by $inc = 1$ marker. The thick curve indicates the average of the statistical power across all burden tests for all genomic regions lengths $\ell \in [2, 4, 6, 8, 10, 12, 14]$. The thin dashed curves represent the power of the burden tests for each length of the associated region separately.

To conclude, we want to stress the fact that burden tests with window-based approaches are very sensitive to inaccuracies due to incomplete or erroneous coverage of the associated regions by the windows. Contrary to this, `FastCMH` successfully circumvents the problem by testing all possible lengths/starting positions of (testable) genomic regions. This fits better the reality of genome-wide association studies, which in practice are often exploratory and little is known about the size and the position of associated genomic regions. The experiments presented above confirm the effectiveness of our method `FastCMH` when compared to window-based burden tests in settings when little is known about the ground truth.

Gene-based burden tests This section provides additional details of the gene-based burden tests that are briefly mentioned in Section 4.1.2 of the manuscript. Here we present simulations to compare `FastCMH` with burden tests models that only test predefined regions of interest. For our simulations, we use the encodings described in Section

S1.5, i.e. Enc.(I) and Enc.(II). Additionally, our method is compared against the two baseline methods FAIS-χ^2 and BonfCMH. The target FWER was set to 0.05 for all approaches. On the y-axis of Figures S20, S21 and S22 we show the evaluation of three statistical metrics: 1) the power, 2) the proportion of confounded regions falsely detected and 3) the FWER, respectively. The x-axis represents the strength of the association ρ between each of the two associated regions and the phenotype. We have $p_s = \rho_{con}$.

Results were averaged over 200 iterations. Burden tests give different results in all seven settings (described in Table S1, Section S1.5.1), because they ignore markers outside genomic windows; this is not the case for the two Tarone-based algorithms nor for BonfCMH. In Figure S20, we observe that the burden tests have a higher power in case (g) than FastCMH and FAIS-χ^2 because exactly all the markers of the associated genomic region are combined in the tested window, while FastCMH performs better in all the other settings. Enc. (II) is slightly more favorable to the burden tests as it sums the single signals, instead of taking the maximum as in Enc. (I), making the combination more robust to noise. Regarding the probability of detecting the confounded genomic region, shown in Figure S21, all conditional tests, except for FAIS-χ^2 that does not condition on confounders, succeed in ignoring the confounded regions. Finally, in Figure S22, we observe that all the Tarone-based methods (FastCMH and FAIS-χ^2) ensure a slightly better control of the FWER than the burden tests do.

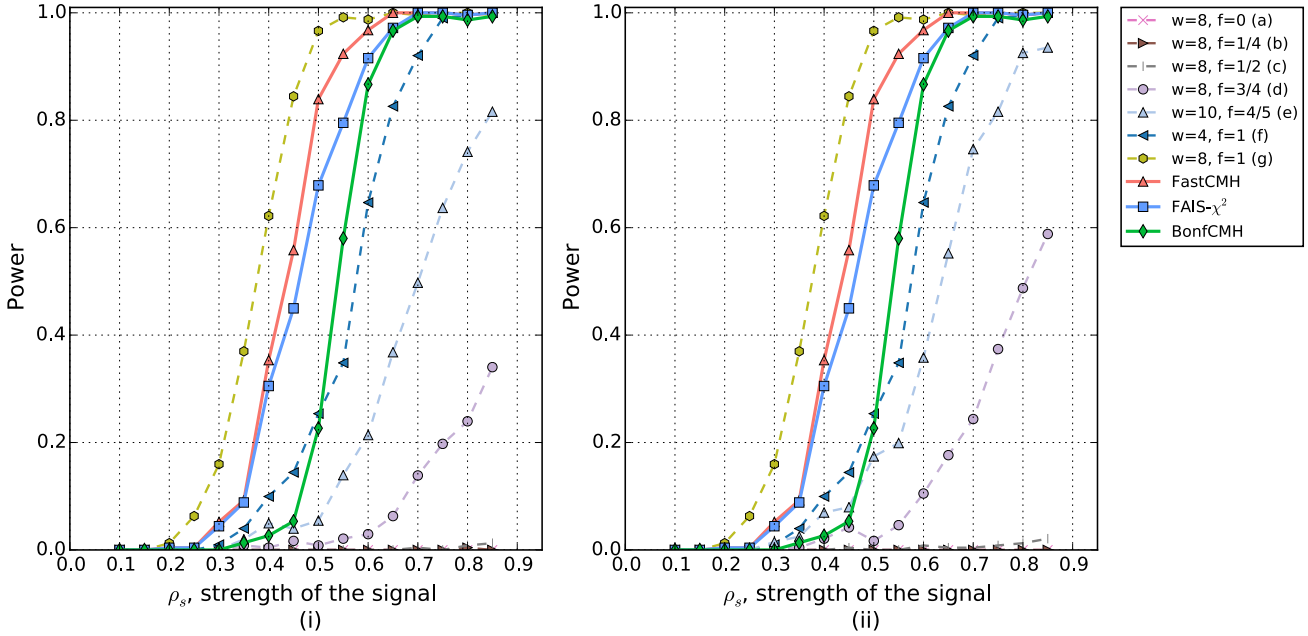

Figure S20: Evaluation of the power as a function of the strength of the signal p_s for FastCMH, FAIS-χ^2, BonfCMH and the burden tests. (i) and (ii) refer to when the encodings Enc.(I) and Enc.(II) are used for the burden tests respectively. The labels (a) to (g) refer to the seven parameter settings in burden tests (see Table S1 in Section S1.5.1), which describe different windows sizes and levels of overlap between the windows of the burden tests and the associated region. The power of burden tests in cases (a), (b) and (c) is close to 0.

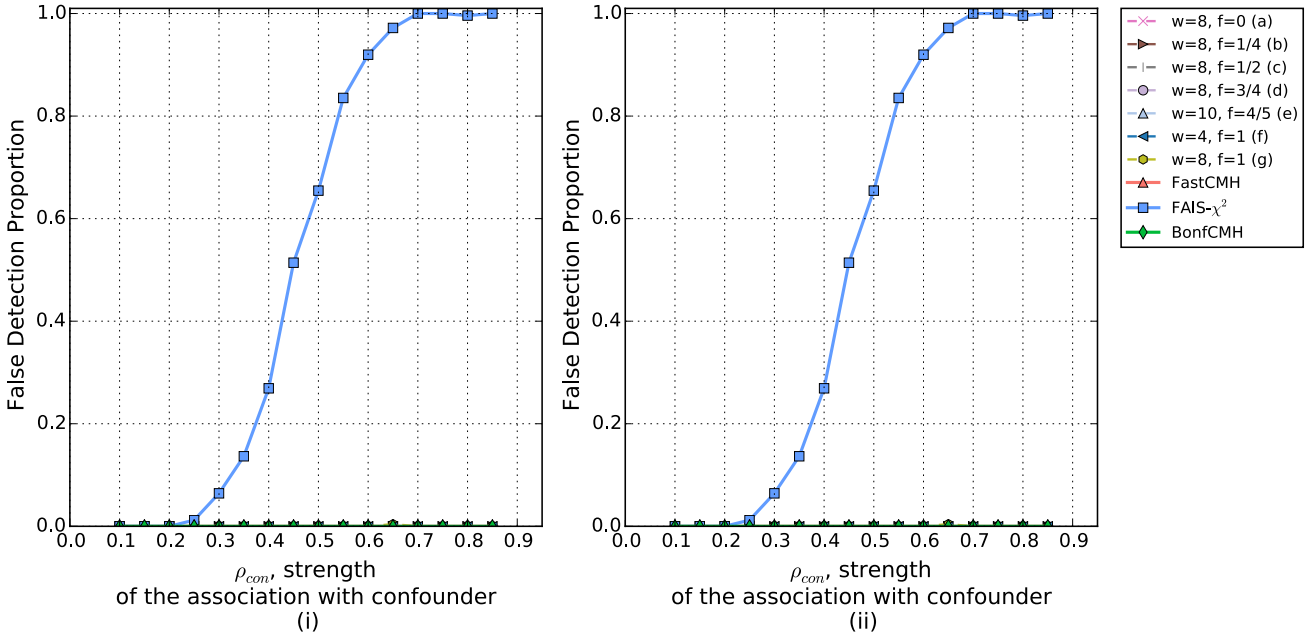

Figure S21: Evaluation of proportion of confounded regions falsely detected as a function of the strength of the association of the confounded interval ρ_{con} for `FastCMH`, `FAIS-`χ^2 , `BonfCMH` and the burden tests. (i) and (ii) refer to when the encodings Enc.(I) and Enc.(II) are used for the burden tests respectively. The labels (a) to (g) refer to the seven parameter settings in burden tests (see Table S1 in Section S1.5.1), which describe different windows sizes and levels of overlap between the windows of the burden tests and the associated region. None of the methods, except for `FAIS-`χ^2, retrieve the confounded genomic region.

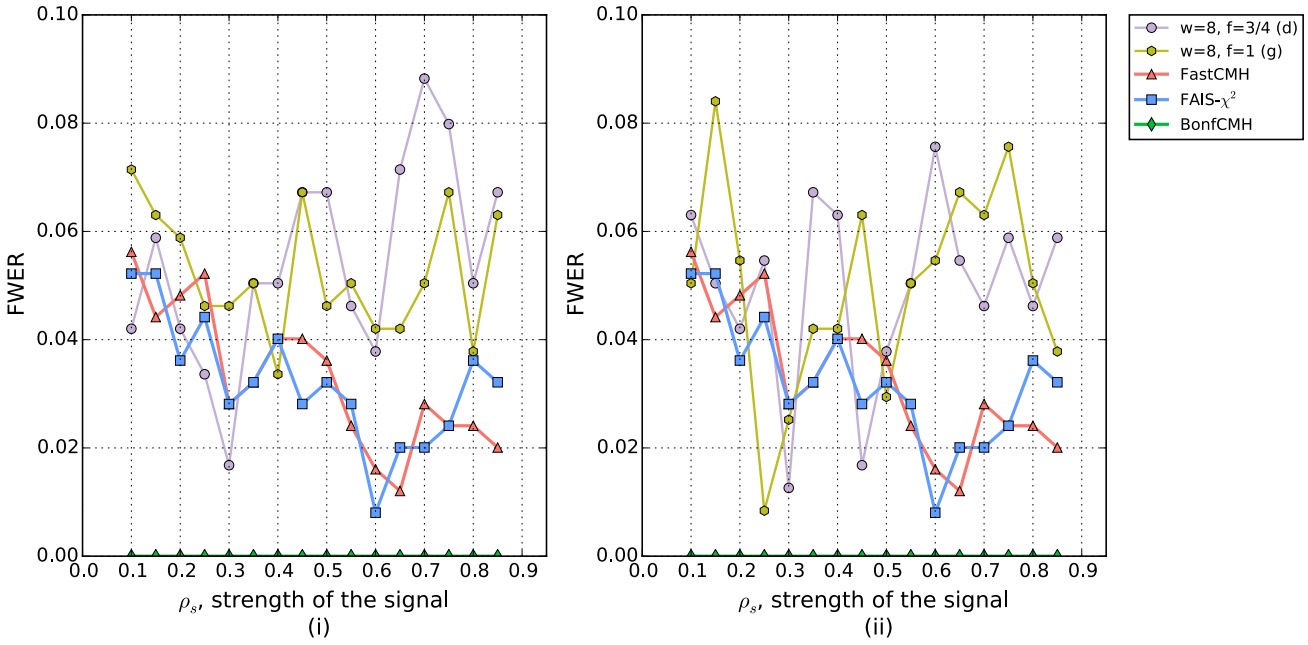

Figure S22: Evaluation of the FWER as a function of the strength of the signal p_s for `FastCMH`, `FAIS-`χ^2, `BonfCMH` and the burden tests. (i) and (ii) refer to when the encodings Enc.(I) and Enc.(II) are used for the burden tests respectively. The labels (d) and (g) refer to two of the seven parameter settings in burden tests (see Table S1 in Section S1.5.1), which describe different windows sizes and levels of overlap between the windows of the burden tests and the associated region. For the sake of clarity, only two burden tests cases are shown, cases (d) and (g). However, the FWER variations for the other burden tests were similar.

In summary, gene-based burden tests exhibit low power and appear to be inefficient at finding genomic regions

that are not almost identical to predefined regions of interest. In contrast, `FastCMH` retrieves associated genomic regions with high power, without the need for predefined biological knowledge to guide the search, while also correcting for confounders.

When combining the simulation results presented for window-based burden tests in addition to the results on gene-based burden tests, it appears that `FastCMH` has superior performance in exploratory genome-wide association studies.

S3.1.6 Testability of genomic regions with respect to their length

As explained in the main manuscript, our definition of meta-marker combined with Tarone's definition of testability, imply that larger genomic regions are less likely to be testable. However, how this dependence exactly manifests strongly depends on factors such as the sample size, the distribution of minor allele frequencies and the amount of linkage disequilibrium between markers.

In this section, we show both the total number and proportion of testable genomic regions as a function of region length for two representative settings of our simulation experiments: (I) a synthetic dataset with no linkage disequilibrium, following the data generation model used for the experiments in the main manuscript as described in Section S2.1.1 and (II) a synthetic dataset with blockwise linkage disequilibrium, following the data generation model described in Section S2.1.2 with $\rho_{ld} \in \{0.25, 0.375, 0.5\}$.

In Figures S23-S26, one can observe that, as expected, both the total number and proportion of genomic regions that are testable decreases as the region size increases. Moreover, there exists a maximum effective region size beyond which all genomic regions are untestable. However, factors such as the existence of linkage disequilibrium can greatly alter the distribution of testable genomic regions: the larger the amount of linkage disequilibrium is, the more genomic regions will be testable and the larger they will be. Indeed, in the case of no linkage disequilibrium, shown in Figure S23, no genomic region with more than 10 markers was testable. In contrast, Figures S24-S26 illustrate how with blockwise linkage disequilibrium, genomic regions with hundreds of markers might be testable. More precisely, with $\rho_{ld} = 0.25$, the maximum effective region size is approximately 100, a value which is roughly doubled for $\rho_{ld} = 0.5$.

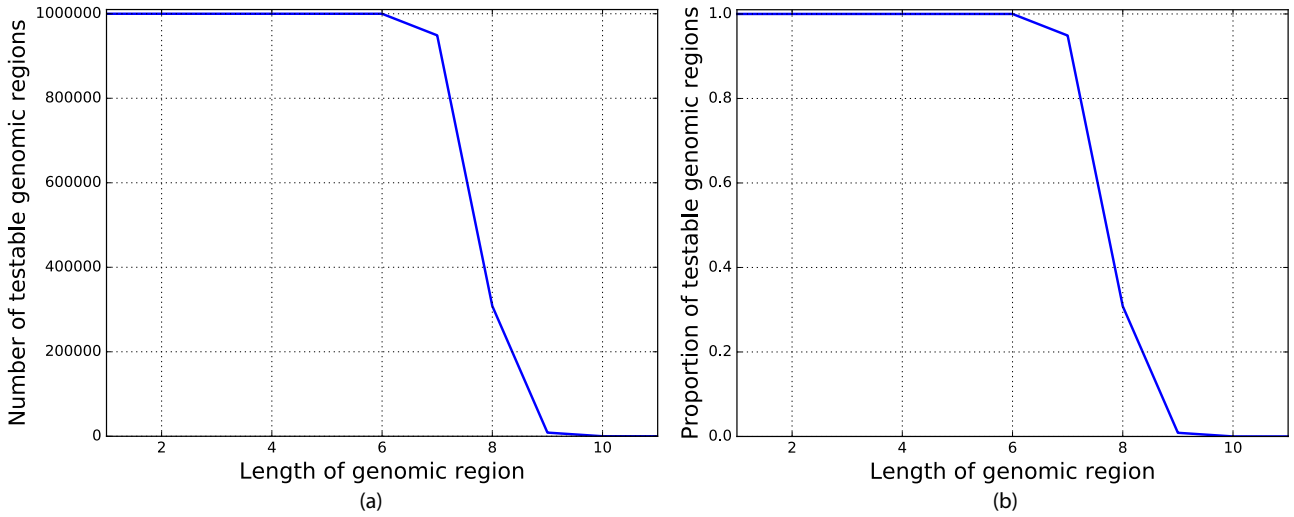

Figure S23: Scenario (I): Model without linkage disequilibrium, as described in Section S2.1.1. (a) Total number of testable genomic regions. (b) Proportion of testable genomic regions.

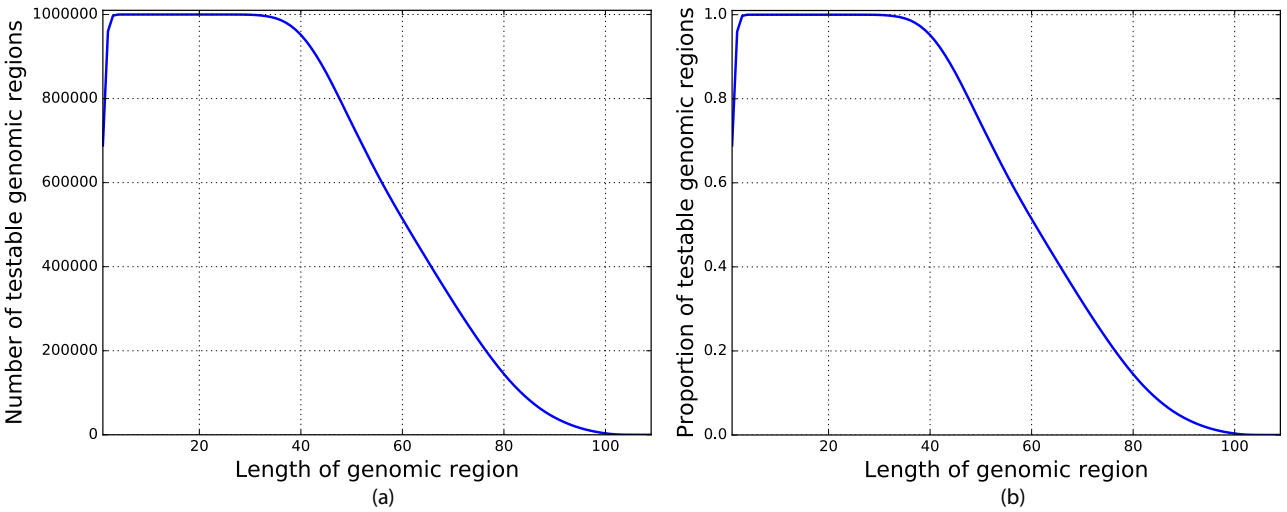

Figure S24: Scenario (II): Model with blockwise linkage disequilibrium ($\rho_{ld} = 0.25$) as described in Section S2.1.2. (a) Total number of testable genomic regions. (b) Proportion of testable genomic regions.

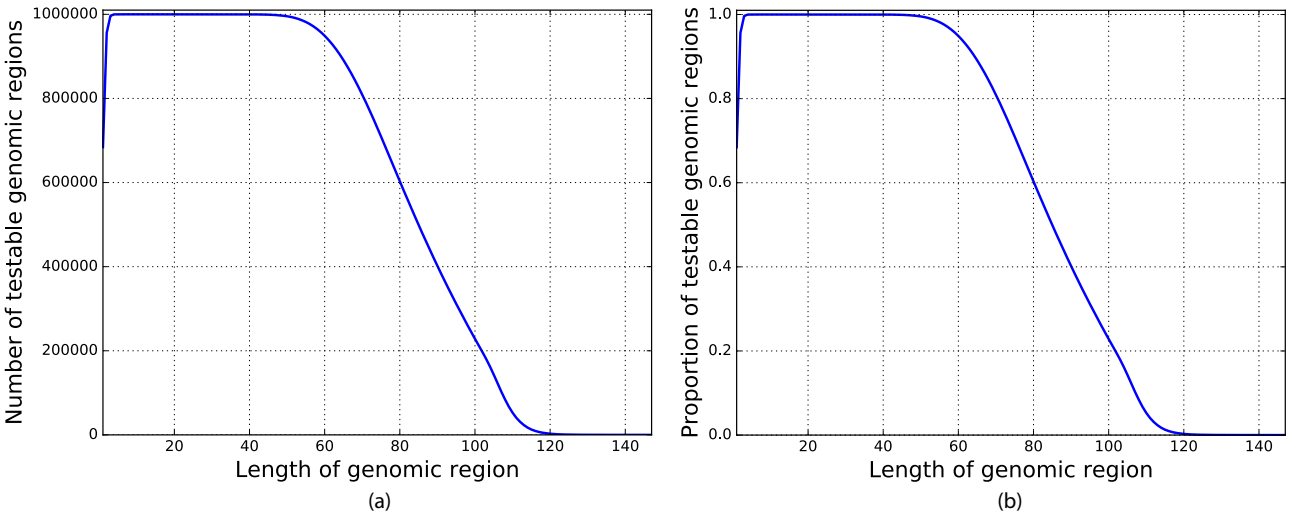

Figure S25: Scenario (II): Model with blockwise linkage disequilibrium ($\rho_{ld} = 0.375$) as described in Section S2.1.2. (a) Total number of testable genomic regions. (b) Proportion of testable genomic regions.

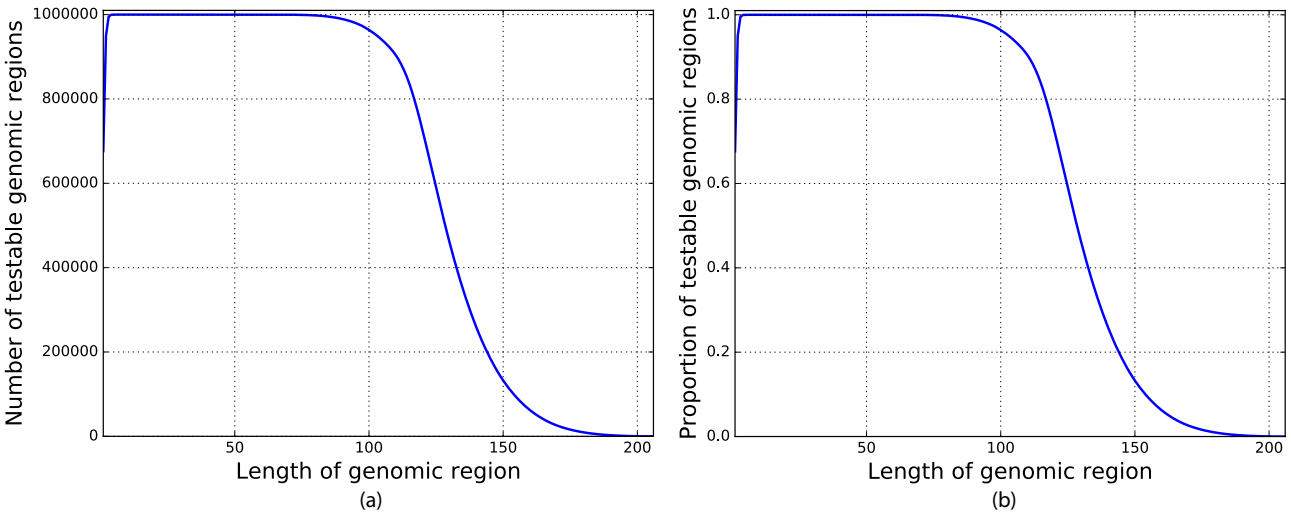

Figure S26: Scenario (II): Model with blockwise linkage disequilibrium ($\rho_{ld} = 0.5$) as described in Section S2.1.2. (a) Total number of testable genomic regions. (b) Proportion of testable genomic regions.

S3.1.7 Comparing `FastCMH` and `FastCMH-FDR`

This simulation study mirrors that in Section 4.1.1 and Figure 2 in the main paper. For $\rho_{\text{s}} = \rho_{\text{con}} \in [0.1, 0.9]$, Figure S27(a) shows that `FastCMH-FDR` has slightly higher power than `FastCMH`, while Figure S27(b) shows that both methods have similar resistance to detecting confounded regions (unlike `FAIS-`χ^2 in Figure 2 in the main paper), although `FastCMH` is slightly better.

However, the increase in power for `FastCMH-FDR` comes at a cost; Figure S27(c) shows the proportion of times that each method will detect at least one false positive (a region which is neither a truly significant region, nor a confounded region), and it can be seen that `FastCMH-FDR` will often detect a false positive, while `FastCMH` will detect a false positive in less 5% of the trials. In fact, we should expect this result for `FastCMH`, since it controls the Family-wise Error Rate at threshold α, and in this simulation we have used $\alpha = 0.05$.

On the other hand, one might be concerned with the regularity with which `FastCMH-FDR` finds a false positive. But this is simply down to the difference between a procedure which controls the FDR and a procedure which controls the FWER; a procedure controlling the FDR will control the *proportion of false discoveries* (false positives) at a threshold α, i.e. a proportion less than α of discoveries will be false, while a procedure controlling the FWER will ensure that there will be at least one false discovery in at most α trials. The two procedures are simply controlling different criteria, and it is up to the practitioner to decide which criterion should be controlled.

Note that there are no runtime comparison figures, because `FastCHM` and `FastCMH-FDR` run in practically the same amount of time. The main computational effort comes from the determination of Tarone's adjusted significance threshold, and this step is common to both methods (Line 1 of Algorithms 1 and 1*). The FWER procedure (`FastCHM`) and the FDR procedure (`FastCMH-FDR`), shown respectively in Line 2 of Algorithms 1 and 1*, are of practically the same complexity, although the FDR procedure requires a sorting procedure. If there are $T = |\mathcal{R}_T(\delta_{tar})|$ testable intervals, then the FWER procedure of `FastCHM` is $O(T)$, while the FDR procedure of `FastCMH-FDR` is $O(T \log T)$.

Also note that it is possible to modify `FastCMH-FDR` to `FastCMH-FDRDep`, which uses $\widetilde{\alpha}$ in Equation (3) instead of α in order to try and account for any positive regression dependence. However, the results for `FastCMH-FDRDep` are not included because, for these parameter settings, they are the practically the same as for `FastCMH-FDR`.

S3.1.8 Fine-grained population structure correction

In this section, we further study the ability of `FastCMH` to correct for population structure when a discretized version of the top principal components of the empirical kinship matrix is used to represent population structure as a categorical covariate with k categories or discretization levels.

To this end, we simulated a dataset that mimics the population structure and degree of confounding of the COPDGene study following the approach described in Song *et al.* (2015), which we summarize here for convenience.

A genotype matrix $\mathbf{G} \in \{0,1\}^{n \times l}$ was generated by sampling the j-th SNP for the i-th individual $\mathbf{g}_i[j]$ as an independent Bernouilli random variable with corresponding minor allele frequence $\mathbf{F}_{i,j}$. The $n \times l$ matrix $\mathbf{F}$ of minor allele frequencies was obtained as $\mathbf{F} = \mathbf{S\Gamma} + f$. In that expression, $\mathbf{S} \in \mathbb{R}^{n \times 2}$ is a matrix containing as columns

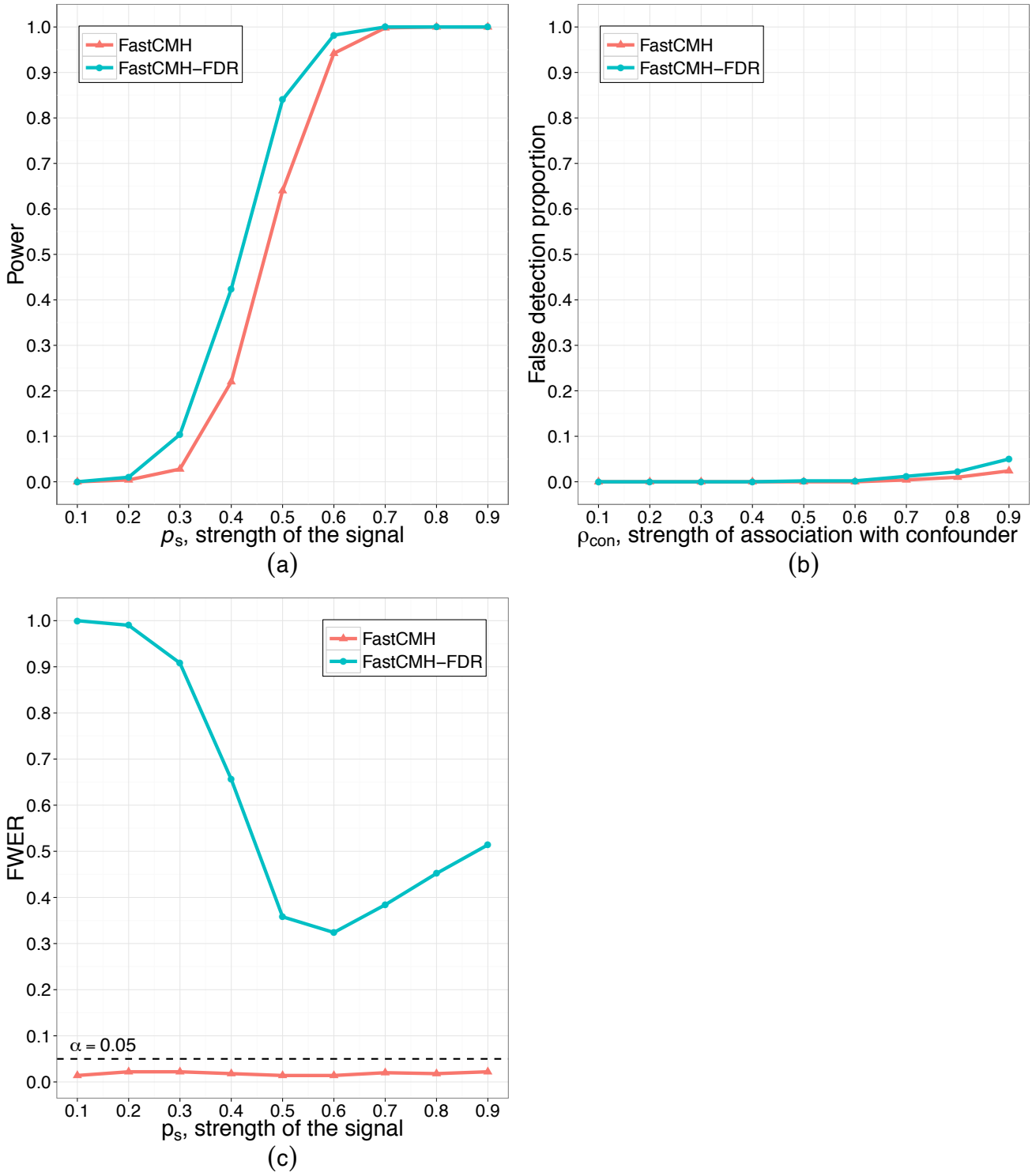

Figure S27: (a) A comparison of the power of `FastCMH` and `FastCMH-FDR` for detecting significant regions as $\rho_s = \rho_{con}$ varies. The parameters are chosen, similar to Figure 2 in the main paper, to be $n = 500, l = 10^5, p_1 = 0.3, k = 2, \rho_s = \rho_{con} \in [0.1, 0.9]$, and 100 trials are used. Besides the in the significant regions, the noise parameter – the probability of a bit being randomly changed – is 0.05. (b) The proportion of confounded significant regions falsely detected by each of the algorithms. The parameters have the same values as for (a). (c) The actual Family-wise error rate of `FastCMH` and `FastCMH-FDR`, where the FWER and FDR thresholds are both set to $\alpha = 0.05$.

the top two principal components of the empirical kinship matrix of the COPDGene study, normalized so that each column takes values in the range $[0, 1]$; $\mathbf{\Gamma} \in \mathbb{R}^{2\times l}$ was sampled as $\Gamma_{l,j} \sim \text{Uniform}(0, 0.45)$ for $l \in \{1, 2\}$; and $f = 0.05$ sets the minimum minor allele frequency in the dataset. The number of individuals n was determined by the number of samples in the COPDGene study, leading to $n = 7,993$. The number of SNPs was set to $l = 100,000$, as was done in Song *et al.* (2015). As a result of this generation process, the population structure in the synthetic genotype matrix $\mathbf{G}$ resembles that of the COPDGene study.

To generate a synthetic case/control phenotype that is confounded by population structure, we followed a procedure based on the generation of confounded quantitative traits in Song *et al.* (2015). More precisely, k-means was used to cluster the individuals into $k^* = 8$ disjoint clusters based on the normalized top two principal components of the original COPDGene study, i.e. the columns of $\mathbf{S}$. Next, for each resulting cluster, synthetic case/control labels y_i were sampled independently for each individual in the cluster as a Bernouilli random variable with success probability equal to the estimated proportion of individuals in the cluster being cases in the original COPDGene study. As a result, the degree of confounding in the synthetic phenotype due to population structure is comparable to that of the original COPDGene study. No true associations between any group of SNPs and the phenotype were included in the generation process. Thus, any findings reported correspond to false positives.

When this simulated dataset is analyzed using `FAIS-`χ^2, $8,501$ significant genomic regions are falsely reported as associated due to its inability to correct for population structure. The resulting genomic inflation was very high, resembling the high-level of inflation in the original COPDGene study ($\lambda = 23.031$).

In practice, we would only have access to $\mathbf{G}$, hence using the columns of $\mathbf{S}$ to represent the structure of the dataset would be over-optimistic. Consequently, to correct for population structure using `FastCMH`, we first computed the top two principal components of the empirical kinship matrix corresponding to $\mathbf{G}$. Note that while those will resemble the original principal components used to generate the synthetic phenotype, i.e. the columns of $\mathbf{S}$, they will not be identical due to sampling variation. To quantize the top two principal components as a categorical covariate with k categories, we employed k-means clustering and set the value of the categorical covariate for each individual to equal the index of the cluster to which the individual is assigned. Again, in practice we would not know what the true number of subpopulations k^* affecting the phenotype is. Thus, we applied k-means with different values of k in the range $[\![2, 15]\!]$ to analyze the ability of `FastCMH` to retrieve k^* and to study its robustness to misspecification in the value of k.

Remarkably, `FastCMH` reported no false positives for any of the 14 different values of k employed in the experiment. Unsurprisingly, the minimum genomic inflation λ was attained when $k = 8 = k^*$ was used as a parameter, leading to $\lambda = 1.018$. However, our results also showed that `FastCMH` is robust to misspecification of k: only the extreme case $k = 2$ led to residual inflation ($\lambda = 1.230$). All other values of k in the range $[\![3, 15]\!]$ led to a satisfactory reduction in genomic inflation, with the resulting λ ranging between 1.018 ($k = 8$) and 1.022 ($k = 6$).

While our results in simulated data illustrate that `FastCMH` can drastically reduce genomic inflation due to complex population structure despite the unconventional need to represent the real-valued principal components of the empirical kinship matrix as a categorical covariate, they also showed that the number of categories k is an important parameter that needs to be considered carefully. While `FastCMH` is designed to handle large values of k in a computationally efficient manner, the sample size can act as a practical limiting factor to the maximum value of k that can be employed. As a consequence, when the sample size is very small, the necessity to use a small value of k might cause some subpopulations to be merged and lead to some residual genomic inflation remaining in the output of `FastCMH`.

S3.2 Results for COPDGene

S3.2.1 Fine-grained population structure correction

As described in Section S2.4, our main experiments for the COPDGene study defined the categorical covariate used by `FastCMH` to correct for confounding by combining ethnicity and height bins. While that lead to an enormous decrease in the inflation factor ($\lambda = 16.70$ before correction, $\lambda = 1.048$ after), it is worth illustrating the possibility of capturing population structure more finely than ethnicity represents. In order to do so, we followed the same approach we employed for analyzing the *A. thaliana* datasets. Firstly, we learn a low-dimensional embedding for each individual, given in our case by the top three principal components of the empirical kinship matrix. Next, we discretize the resulting embedding using k-means. The category for each individual is given by the ID of the cluster they are assigned to.

We applied that procedure for values of k in the range $k \in [\![2, 10]\!]$. We observed the following: (i) regardless of the value of k, this approach retrieves the same three significant genomic regions as the approach described in the main manuscript; (ii) using $k = 2$ is virtually identical to using ethnicity as the covariate, leading to $\lambda = 1.048$; (iii) the lowest genomic inflation factor was achieved for $k \in \{3, 6, 7, 8, 9, 10\}$, leading to λ ranging between 1.011 ($k = 8$) and 1.016 ($k = 3$). All those points together illustrate that many values of k in `FastCMH` might lead to satisfactory performance, showing a certain robustness to the choice of parameters.

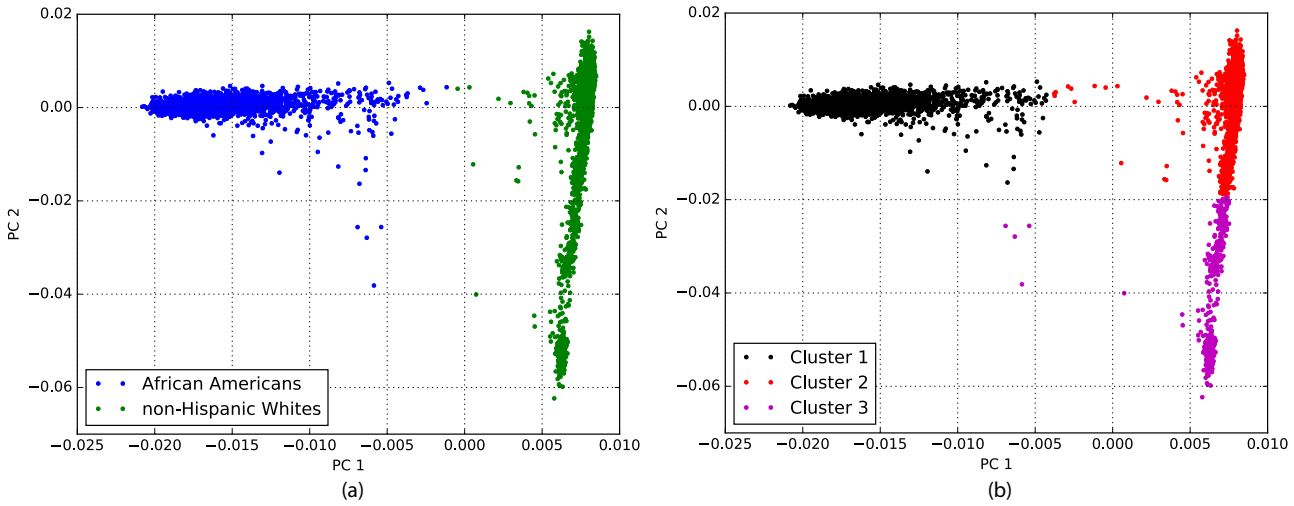

Figure S28: Embedding of all 7,993 samples in the COPDGene study according to the two principal components of the kinship matrix: (a) Individuals colored according to ethnicity. (b) Individuals colored according to the category assigned by k-means clustering ($k = 3$) on the three principal components of the kinship matrix.

In Figure S28 we show a 2D embedding of the individuals using the top two principal components of the kinship matrix, colored according to ethnicity (a), and the categories obtained using k-kmeans with $k = 3$ (b). In essence, k-means splits the non-Hispanic White individuals into two subclusters, hinting at the possibility of residual population structure within the two original subgroups. The resulting QQ-plot is depicted in Figure S29. As it can be seen, this fine-grained definition of the covariate results in slightly better agreement with the expected distribution of p-values, consistent with the decrease in genomic inflation.

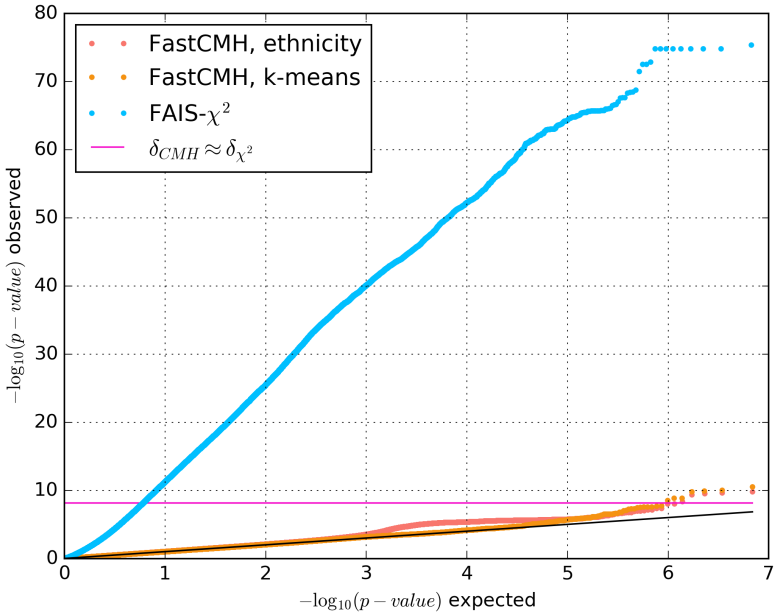

Figure S29: QQ-plot for the p-values of all testable genomic regions obtained with `FastCMH` (red) and `FAIS`-χ^2 (blue) for the COPDGene study. The categorical covariate was obtained by applying k-means clustering with $k = 3$ on the top three principal components of the kinship matrix.

Influence of the number of principal components used to represent population structure

A popular approach to account for population structure is to include the top p principal components of the empirical kinship matrix as additional covariate factors in a regression model (Price *et al.*, 2006). Increasing the number p of top principal components used as covariates generally leads to better population structure correction. However, beyond a certain value of p, the reduction in inflation due to including additional principal components as covariates becomes marginal. This leads to a natural way to select the value of parameter p. When using principal components of the kinship matrix to represent population structure, `FastCMH` requires an additional parameter k representing the number of categories in which the resulting p-dimensional embeddings for each individual are discretized. Intuitively, the number p of principal components controls the amount of information about the population structure that is included in the model prior to discretization, while the number of categories for the covariate k controls how finely that information is represented in the categorical covariate.

While the experiments described above used $p = 3$, chosen according to the criterion just described, in the remaining of this section we explore the robustness of `FastCMH` to the joint effect of parameters p and k. To that end, we analyzed the resulting genomic inflation in the COPDGene study after applying `FastCMH` using a categorical covariate obtained by discretizing the top p principal components of the empirical kinship matrix into k categories using unweighted k-means. The parameters p and k were both varied independently in the range $[\![2, 10]\!]$, resulting in 81 different experiments each corresponding to a different combination of values for p and k.

The results shown in Figure S30 illustrate that many different choices for p and k lead to a satisfactory reduction of genomic inflation. In particular: (I) the median genomic inflation across all 81 parameter combinations is $\lambda = 1.020$; (II) for any fixed value of the number p of top principal components in the range $[\![2, 10]\!]$, the best performing choice of the number k of categories achieves λ in the range $[1.009, 1.020]$; (III) for any fixed value of the number k of categories in the range $[\![3, 10]\!]$, the best performing choice of the number p of top principal components achieves λ in the range $[1.009, 1.014]$; and (IV) for $k = 2$ categories, any choice of the number p of principal components in the range $[\![2, 10]\!]$ recovers the original ethnicity, leading to a genomic inflation of $\lambda = 1.048$. Most interestingly, 88.89% of all 72 parameter combinations with $k \geq 3$ categories achieved better population structure correction than using ethnicity alone. Reassuringly, the three significantly associated genomic regions found in the original analysis remain significant for all 81 parameter combinations in this new analysis.

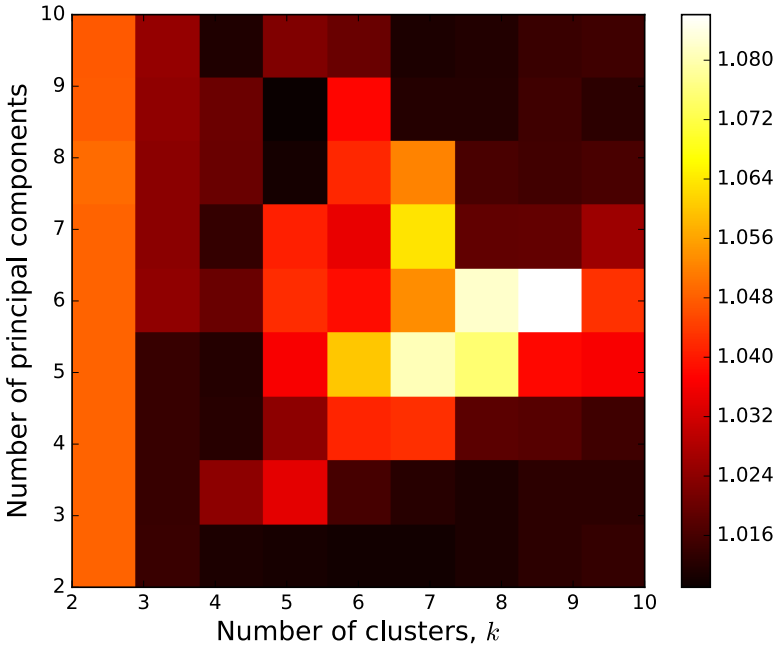

Figure S30: Genomic inflation λ after applying `FastCMH` to the COPDGene study for different values of p, the number of top principal components of the empirical kinship matrix included in the analysis, and k, the number of categories of the covariate. Both parameters were varied independently in the range $[\![2, 10]\!]$.

Most combinations of p and k lead to a considerable reduction in genomic inflation and, for any given fixed choice of p or k, there exists at least one value of the other parameter that leads to near-optimal performance. However, Figure S30 shows that `FastCMH` is not entirely insensitive to the choice of parameters, with a minority of parameter combinations leading to a suboptimal reduction in genomic inflation (e.g. $\lambda = 1.086$ for $p = 6$ and $k = 9$). Thus, if one wishes to use `FastCMH` to correct for population structure using principal components of the empirical kinship matrix, it is recommended to explore multiple combinations of p and k for better performance.

More generally, these experiments illustrate that the way real-valued covariates are discretized into a categorical covariate for use in `FastCMH` might have an influence in the ability of our approach to reduce genomic inflation. In particular, in cases for which the number of real-valued covariate factors to correct for is large, such as when one wishes to include a large number p of principal components of the empirical kinship matrix as covariate factors, it might become difficult to find an appropriate discretization of the set of continuous covariates using clustering algorithms as simple as unweighted k-means. Therefore, exploring more sophisticated ways to represent a set of real-valued covariates as a categorical variable is a relevant topic for future work that might eventually lead to improved performance in `FastCMH`.

S3.2.2 Impact of the number of categories for the covariates on the runtime

Figure S31 illustrates how the number k of categories for the covariate affects the runtime in the COPDGene study. The covariates were defined as in Section S2.4. However, varying values of k were obtained by changing the number of categories in which the covariate "height" was discretized. It is clear from the figure that the search space pruning algorithm is very efficient in terms of running time. This is in contrast to a naive implementation of Tarone's trick for CMH, whose runtime scales exponentially with k.

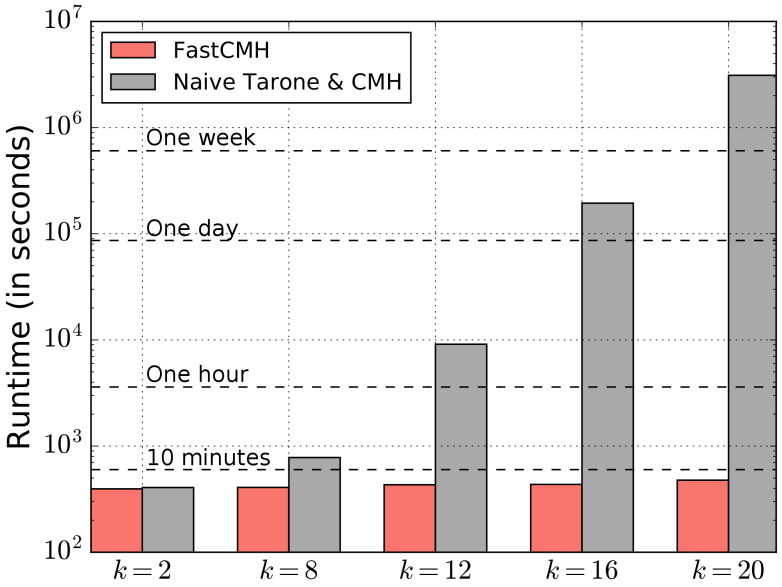

Figure S31: Runtime of `FastCMH` and a naive implementation of Tarone's trick for the CMH test in the COPDGene study.

S3.2.3 Significant genomic regions

The three trait-related and statistically significant genomic regions reported by `FastCMH` overlap with the (CHRNA5-CHRNA3-CHRNB4) nicotinic acetylcholine receptor gene cluster. Each region and the SNPs within it are listed in Table S4. Some of the SNPs (e.g. rs6495306) have actually been reported to lack evidence of association with COPD in a GWAS dataset (Qiu *et al.*, 2011). Of course, our results do not contradict these previous studies but show the potential of our method to uncover genomic regions that are associated to a phenotype of interest for which traditional GWASs lack the statistical power to detect.

The table shows the p-values attained by the region as well as the p-values of each individual SNP, computed as a single SNP association test. This is to highlight that even when individual SNPs have a weak association to the phenotype, their cumulative effect within a region can attain statistical significance.

Table S4: Details of the statistically significant genomic regions reported by `FastCMH` and of the SNPs contained in them.

Genomic region	Gene overlap	p-values `FastCMH`	
		region	single SNP
chr15 78863472–78865893	CHRNA5	2.03e-10	
rs667282			9.06e-08
rs6495306			4.60e-02
chr15 78907656–78909480	CHRNA3	4.54e-10	
rs6495308			3.55e-05
rs12443170			2.96e-03
rs3743074			4.30e-02
chr15 78917399–78928264	CHRNB4	1.41e-10	
rs1948			1.00e-02
rs950776			6.18e-02
rs12441088			1.70e-05

Corrected significance threshold for:

- all testable intervals: 7.26e-09
- all testable single SNPs: 8.12e-08

S3.2.4 Analysis of individual cohorts vs. merged dataset with both cohorts

We decided to evaluate the following two scenarios:

- run FAIS-χ^2 individually on the African American (AA) cohort and non-Hispanic whites (NHW) cohort.
- run FastCMH on the merged dataset of AA+NHW. This is the analysis that was conducted in the main paper.

The genomic inflation factor evaluated on the p-values of all testable regions of FAIS-χ^2 is very low and essentially equal to that attained by FastCMH on the merged dataset. This serves as additional evidence that: i) the most important confounding factor in this dataset is the "population ID" and ii) FastCMH correctly corrects the inflation.

When FAIS-χ^2 was run individually on each population, no regions were found for AA and only one region was found for NHW. Thus, if we had computed the significant intervals as the intersection of those reported on each dataset individually, nothing would have been found. Even if we had taken the union instead of the intersection, only one significant interval would have been found. The p-values for each of the three significant intervals reported by FastCMH on the entire dataset are shown in Table S5.

Table S5: Comparison of FAIS-χ^2 when run individually on the African American (AA) and non-Hispanic white (NHW) populations. The three genomic regions are the ones reported by FastCMH and discussed in the main paper.

Genomic region	FAIS-χ^2		FastCMH
	AA	NHW	AA+NHW
chr15 78863472–78865893	7.52e-5	6.12e-7	2.22e-10
chr15 78907656–78909480	1.08e-1	5.88e-10	3.25e-10
chr15 78917399–78928264	5.82e-4	5.89e-8	1.77e-4

The significance thresholds are:

- for AA = 8.14e-9
- for NHW = 8.71e-9
- for AA+NHW (merged) = 7.25e-9

S3.2.5 Burden tests

For the COPDGene study, when performing the gene-based burden tests as described Section S1.5.2, *none* of the three genes (CHRNA5-CHRNA3-CHRNB4) found by FastCMH were significant using *any* of the burden tests. When taking the smallest p-value across all burden tests performed, only CHRNB4 was close to significance (p-value 5.72e-6) while CHRNA5 and CHRNA3 had p-values 0.24 and 0.41, respectively. While each of the three significantly associated genomic regions found by FastCMH overlaps with one gene in the cluster (CHRNA5-CHRNA3-CHRNB4), the significant regions do *not* span the entire gene. Burden tests, which do not consider sub-regions, include too many markers in the test, diluting the signal among noise and missing the association. In contrast, gene-based burden tests identified the gene ZRANB3 as significant, with the smallest p-value across all burden tests being 1.56e-6. FastCMH assigns the genomic region corresponding to ZRANB3 a very similar p-value, 2.31e-6. However, ZRANB3 is not significantly associated for FastCMH because it uses a more stringent significance threshold. This behavior is to be expected, as there are many more testable genomic regions ($\approx 7 \cdot 10^6$) than genes ($\approx 1.7 \cdot 10^3$) in the COPDGene dataset.

When the window-based burden tests were conducted on the two window sizes used to partition the genome (500 kilobases and 1 megabase) as described Section S1.5.2, the results coincided with the findings of the gene-based tests (i.e., overlap with the gene ZRANB3). For 500 kb windows, only the region chr2:136,018,946:136,518,946 is found when the encoding is (I). Likewise, for 1 megabase windows, the region chr2:136,018,810:137,018,810 is found when the encoding used is (II).

S3.2.6 Testability of genomic regions with respect to their length

In this section, we show the total number and proportion of genomic regions that were testable in the COPDGene study. As it can be seen from Figure S32, the largest testable genomic regions contained approximately 120 markers, with most regions having 20 markers or less.

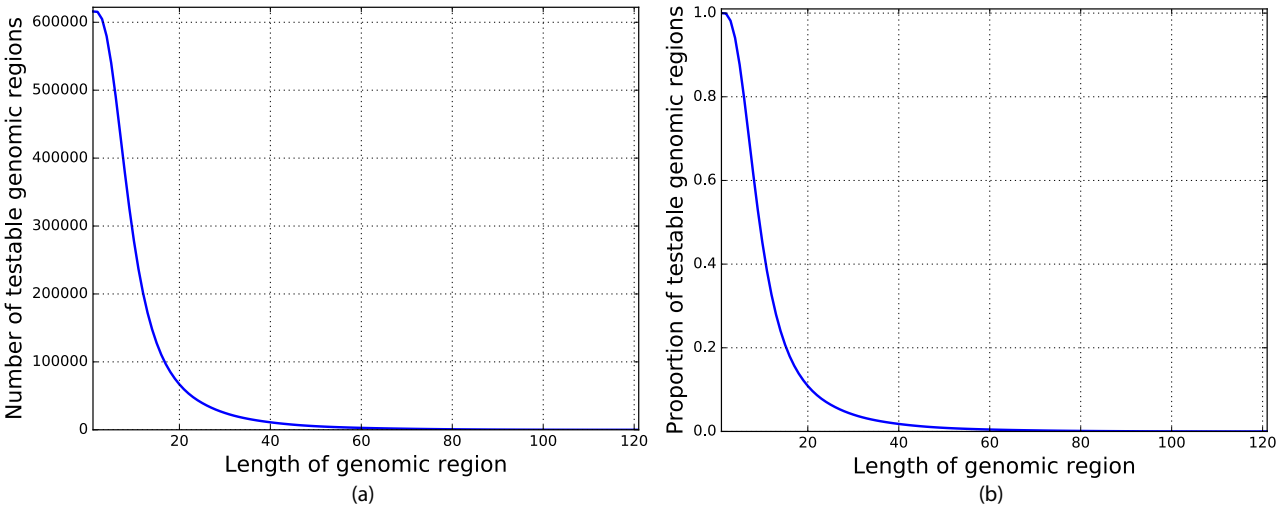

Figure S32: COPDGene study: (a) Total number of testable genomic regions. (b) Proportion of testable genomic regions.

S3.2.7 Comparing `FastCMH` and `FastCMH-FDR`

To see how `FastCMH-FDR` compares with `FastCMH` in practice, we ran two versions of `FastCMH-FDR` on the COPDGene data set: (i) `FastCMH-FDR`, which uses the original version of FDR Benjamini and Hochberg (1995b), which does not take any dependence into account, and (ii) `FastCMH-FDRDep`, which uses the version of FDR that takes positive regression dependence into account Benjamini and Yekutieli (2001) (see Section S1.4 and Equation (3)).

Now, recall that `FastCMH` finds 3 significant regions while `FAIS-`χ^2 finds 88, 403 significant regions. `FastCMH-FDRDep` discovers the same 3 significant regions as `FastCMH`. However, `FastCMH-FDR` finds 70 regions.

The reason that `FastCMH-FDR` discovers additional regions is that the FDR procedure allows regions with p-values as high as 2.09e-5 to be declared significant. On the other hand, the adjusted threshold of `FastCMH` is 7.25e-9, so only regions with p-values smaller than 7.25e-9 can be declared significant by `FastCMH`. Figure S27 shows that while `FastCMH-FDR` has higher power than `FastCMH` (Figure S27(a)), it comes at the cost of a high FWER, while `FastCMH` preserves the FWER at level $\alpha = 0.05$.

Amongst the 70 regions discovered by `FastCMH-FDR` are two regions which overlap with the ZRANB3 gene. Recall from Section S3.2.5 that the burden test discovered a significant region which overlapped with ZRANB3. So, although `FastCMH` did not find this genetic region which was declared to be significant by the burden test, `FastCMH-FDR` did find this region.

S3.3 Results for *Arabidopsis thaliana*

S3.3.1 Additional QQ-plots

The QQ-plots shown in Figures S33–S35 correspond to the datasets *avrB*, *avrRphB* and *avrPpm1* respectively. In all cases, `FastCMH` considerably decreases the level of inflation present in the results, compared to those obtained with its predecessor `FAIS`.

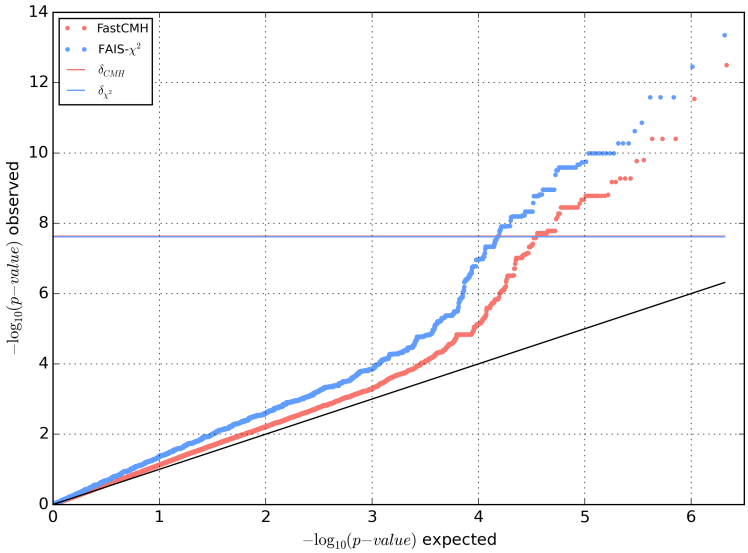

Figure S33: QQ-plot for the p-values of all testable genomic regions of the *avrB* dataset, for both `FastCMH` (red) and `FAIS`-χ^2 (blue) algorithms.

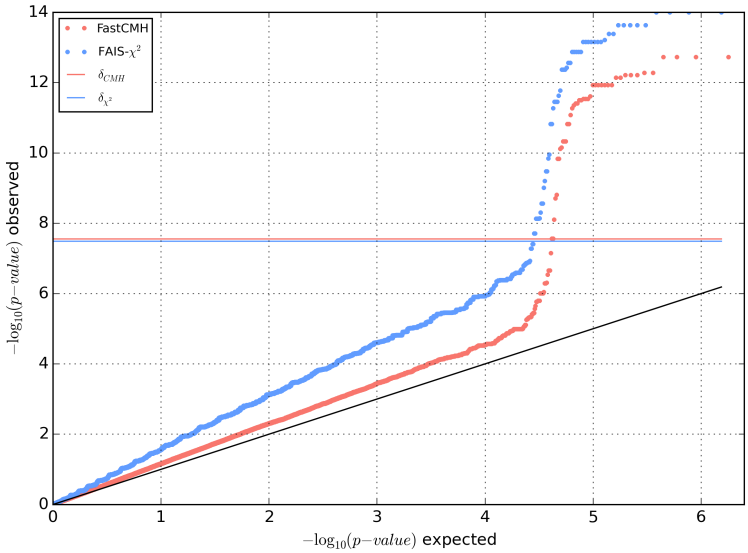

Figure S34: QQ-plot for the p-values of all testable genomic regions of the *avrRphB* dataset, for both `FastCMH` (red) and `FAIS`-χ^2 (blue) algorithms.

S3.3.2 Significant genomic regions

The most significant genomic regions found by `FastCMH` in *A. thaliana* are shown in Table S6.

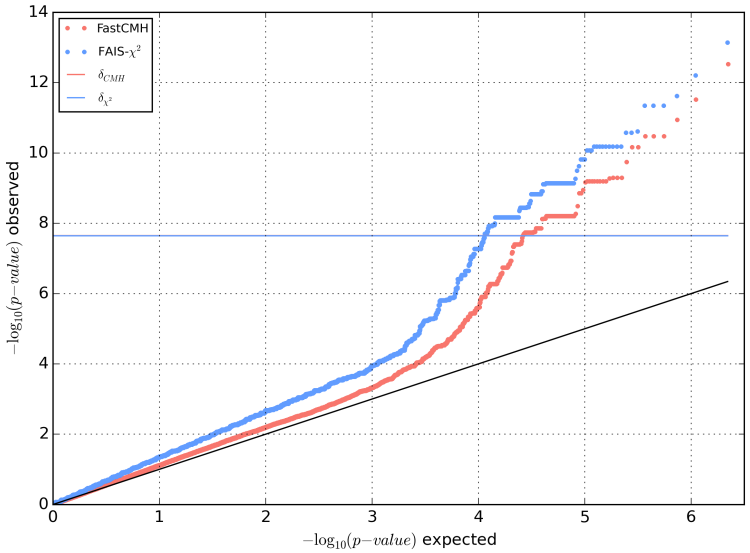

Figure S35: QQ-plot for the p-values of all testable genomic regions of the *avrRpm1* dataset, for both `FastCMH` (red) and `FAIS`-χ^2 (blue) algorithms.

Table S6: Details of the most statistically significant genomic regions reported by `FastCMH`. Underlined SNPs are contained in genes (including markers at a distance smaller than 10 kb). The SNP notation in the format: *Chr*4_2398754 indicates a SNP located on the 4^{th} chromosome at the position 2398754.

SNPs in the significant genomic region	Gene overlap	p-values FastCMH
avrB		
▷ *Chr*3_2225653 - *Chr*3_2225893	NA	3.15e-13
▷ *Chr*3_2221399 - *Chr*3_2222856	AT3G07020	1.61e-10
▷ *Chr*3_2227817	AT3G07040	1.69e-10
▷ *Chr*3_2288913 - *Chr*3_2289178 - *Chr*3_2289559	AT3G07195	5.34e-10
avrRpm1		
▷ *Chr*3_2225653 - *Chr*3_2225893	NA	3.00e-13
▷ *Chr*3_2227817	AT3G07040	1.15e-11
▷ *Chr*3_2310055 - *Chr*3_2311035 - *Chr*3_2311574	AT3G07260	6.90e-11
avrPphB		
▷ *Chr*1_4146714	AT1G12220	1.87e-13
▷ *Chr*1_4143163	AT1G12210	1.87e-13
▷ *Chr*1_4141624	AT1G12210	1.17e-12
▷ *Chr*1_4139802 - *Chr*1_4140044	AT1G12200	2.43e-12
LES		
▷ *Chr*4_8297892	AT4G14400	1.37e-09
▷ *Chr*5_6485290	NA	7.39e-09
▷ *Chr*4_8307440 - *Chr*4_8307761 - *Chr*4_8307910 - *Chr*4_8308076 - *Chr*4_8308306 - *Chr*4_8308768 - *Chr*4_8308977	AT4G14440	2.25e-08
LY		
▷ *Chr*5_18925351	AT5G46640	1.27e-08

S3.3.3 Burden Tests

In Table S8, we synthesize the results obtained with the different versions of the gene-based burden tests and for `FastCMH`. For all gene-based burden tests, the significance threshold is obtained using Bonferroni's correction, resulting in a value of 2.15e-06. When comparing the results of `FastCMH` with those of the gene-based burden tests, if multiple intervals that were deemed significantly associated by `FastCMH` overlap with a gene, the p-value chosen is the smallest one. Table S7 displays all the genomic inflation factors across gene-based burden tests and `FastCMH`. We note that across all phenotypes and burden tests, *dummy - (II)* and *PCs - (II)* burden tests always display the highest genomic inflation factors and a higher number of significant genes. For the three phenotypes, *avrB, avrRpm1* and *avrPphB*, we note that `FastCMH` retrieves more significant genes and genomic regions than

the gene-based burden tests although the genomic inflation is in the same range. For LES and LY, the genomic inflation factors for `FastCMH` and CMH are the smallest across all and, as a consequence, those two tests retrieve less significant genomic regions or genes. These results show that the two methods are complementary regarding gene discovery. However, `FastCMH` is also able to find SNPs or genomic regions that are not inside genes, as partially shown by Table S6. In total, out of all SNPs found by `FastCMH`, 45% are not inside genes. Moreover, 33% of significant genomic regions found by `FastCMH` are partially outside genes and 9% are fully outside genes. The variability across burden tests and the complementarity of `FastCMH` is also illustrated in Figure S36.

When the window-based burden tests were conducted on the two window sizes used to partition the genome (500 kilobases and 1 megabase), we only find hits with the encoding Enc.(II) and with three phenotypes out of five. For phenotype LY, the region 'Chr1:9,000,019:9,500,019' is significant for the window size 500kb; for phenotype *avrB*, the regions 'Chr1:5,000,006:6,000,006' and 'Chr5:1,000,002:2,000,002' are significant for the window 1000kb size and the region 'Chr1:9'000'019:9'500'019' is significant for the window size 500kb; for the phenotype avrRpm1, the region 'Chr1:5,500,012:6,000,012' is significant for the window size 500kb and the region 'Chr1:5,000,006:6,000,006' is significant for the window size 1000kb. We notice that the results of the gene-based burden tests are complementary to the ones obtained by `FastCMH` and by the gene-based tests, which shows again that burden tests' results have a large variability that depends on the definition of the genomic regions being tested and on the encodings.

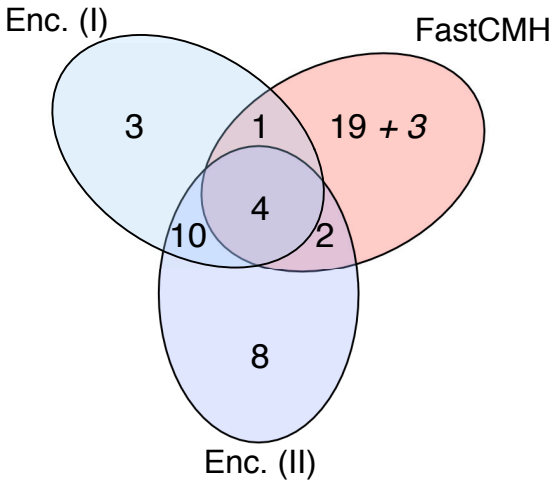

Figure S36: Venn diagram for the genes found by the gene-based burden tests and/or `FastCMH`. Keys for the legend: Enc.(I) represents the union between all the gene-based burden tests that use encoding (I) as described in Section S1.5 encode the meta-marker with an OR (including CMH) ; Enc. (II) represents the union of all the gene-based burden tests that use encoding (II) for the meta-marker (see Section S1.5); `FastCMH` considers both: genes and intervals fully outside genes (in italic) that are only retrieved by `FastCMH`. We note that this figure is conservative as we are taking the union of gene-based burden tests as comparison partners.

Table S7: Genomic inflation factors for all gene-based burden tests and for `FastCMH` across all phenotypes. Note that for `FastCMH`, the genomic inflation factor is calculated using p-values for testable genomic regions only, leading to an inflated genomic inflation factor. Keys to abbreviations: *dummy* indicates that the covariates are coded as k dummy indicator variables, *PCs* means that we chose the three first principal components of the kinship matrix as covariates, *(I)* and *(II)* correspond to the encodings described in Section S1.5. Finally, *CMH* corresponds to the gene-based burden test using the CMH test applied to encoding (I) for each gene.

Phenotype	`FastCMH`	Burden tests				
		dummy - (I)	*dummy - (II)*	*PCs - (I)*	*PCs - (II)*	*CMH*
avrB	1.17	1.12	1.22	1.07	1.17	1.05
avrRpm1	1.13	1.13	1.24	1.07	1.15	1.03
avrPphB	1.22	1.41	1.62	1.15	1.27	1.12
LES	1.21	1.43	1.68	1.23	1.43	1.16
LY	1.30	1.44	1.63	1.44	1.63	1.20

Table S8: The statistically significant genomic regions reported by the different gene-based burden tests (resp. `FastCMH`) and the corresponding gene (resp. genomic region) p-values when significant. A — indicates that the gene is not significant for the given test. In bold, we indicate genes that are found by `FastCMH`. Keys to abbreviations: *dummy* indicates that the covariates are coded as k dummy indicator variables, *PCs* means that we chose the three first principal components of the kinship matrix as covariates, *(I)* and *(II)* correspond to the encodings described in Section S1.5. Finally, *CMH* corresponds to the burden test using the CMH test applied to encoding (I) for each gene.

Significant gene	Number of SNPs	`FastCMH`	Burden tests and p-values				
			dummy - (I)	*dummy - (II)*	*PCs - (I)*	*PCs - (II)*	*CMH*
avrB							
▷ **AT3G07050**	10	1.66e-09	—	1.01e-06	—	1.76e-07	—
▷ **AT3G07195**	4	5.34e-10	1.23e-10	6.45e-10	2.01e-11	1.78e-10	2.77e-10
▷ **AT3G07010**	10	5.24e-09	1.44e-06	—	—	—	—
▷ **AT3G07020**	—	1.61e-10	—	—	—	—	—
▷ **AT3G07040**	—	1.69e-10	—	—	—	—	—
▷ **AT3G07060**	—	3.58e-09	—	—	—	—	—
▷ **AT3G07070**	—	3.58e-09	—	—	—	—	—
▷ **AT3G07260**	—	6.68e-10	—	—	—	—	—
▷ **AT3G07330**	—	2.12e-09	—	—	—	—	—
avrRpm1							
▷ **AT3G07050**	10	6.42e-10	—	4.04e-07	—	8.75e-08	—
▷ **AT3G07195**	4	5.06e-10	4.04e-10	1.07-10	5.11e-11	1.39e-11	5.07e-10
▷ AT3G07005	10	—	7.20e-07	7.20e-07	—	—	—
▷ **AT3G07020**	—	1.81e-10	—	—	—	—	—
▷ **AT3G07040**	—	1.15e-11	—	—	—	—	—
▷ **AT3G07060**	—	6.26e-09	—	—	—	—	—
▷ **AT3G07070**	—	6.26e-09	—	—	—	—	—
▷ **AT3G07200**	—	1.77e-08	—	—	—	—	—
▷ **AT3G07250**	—	1.77e-08	—	—	—	—	—
▷ **AT3G07260**	—	6.90e-11	—	—	—	—	—
▷ **AT3G07330**	—	7.5e-09	—	—	—	—	—
avrPphB							
▷ **AT1G12210**	9	1.87e-13	1.67e-06	1.18e-15	6.02e-08	7.94e-20	—
▷ **AT1G12220**	3	1.87e-13	3.92e-14	6.18e-16	3.25e-17	2.43e-19	6.12e-13
▷ AT1G12230	3	—	7.78e-14	3.83e-15	2.76e-14	1.57e-16	3.19e-12
▷ AT5G11340	5	—	—	—	—	8.91e-07	—
▷ AT5G11350	3	—	1.08e-06	—	4.77e-08	1.58e-07	9.94e-07
▷ AT1G12170	5	—	—	—	7.89e-07	—	—
▷ **AT1G12200**	—	2.43e-12	—	—	—	—	—
▷ **AT1G12190**	—	7.95e-09	—	—	—	—	—
LES							
▷ AT3G06120	3	—	—	2.01e-07	—	—	—
▷ AT4G28890	7	—	4.38e-07	4.28e-07	1.77e-07	1.77e-07	—
▷ AT4G14410	9	—	—	—	—	2.74e-07	—
▷ AT1G34420	3	—	—	—	—	1.87e-06	—
▷ AT1G08500	2	—	—	—	1.28e-07	1.28e-07	—
▷ AT5G45780	6	—	—	—	3.99e-07	2.45e-07	—
▷ AT3G18535	5	—	—	—	—	—	1.25e-06
▷ AT4G39955	7	—	—	—	—	—	4.36e-07
▷ **AT4G14440**	—	2.25e-08	—	—	—	—	—
▷ **AT4G14400**	—	1.37e-09	—	—	—	—	—
LY							
▷ AT1G34420	3	—	—	1.21e-07	—	—	—
▷ AT2G38995	8	—	—	1.63e-06	—	—	—
▷ AT3G61480	5	—	1.57e-06	1.57e-06	—	—	—
▷ AT5G46660	5	—	—	1.99e-06	—	5.06e-07	—
▷ AT5G49620	1	—	1.61e-06	1.61e-06	—	—	—
▷ AT5G45780	6	—	—	—	—	1.49e-06	—
▷ AT1G08500	2	—	—	—	3.36e-08	3.36e-08	—
▷ AT2G18120	3	—	—	2.08e-07	1.87e-06	—	—
▷ **AT5G46640**	—	1.27e-08	—	—	—	—	—

S3.3.4 Testability of genomic regions with respect to their length

In this section, we show the total number and proportion of genomic regions that were testable for all five *Arabidopsis thaliana* datasets. In Figures S37-S41, we observe the same trend as in our synthetic datasets (see Section S3.1.6) and the COPDGene study (see Section S3.2.6): both the total number and the proportion of testable genomic regions decrease with the number of markers in the region for all five *A. thaliana* datasets. However, compared to the results in the COPDGene study, the proportion of testable genomic regions if significantly smaller in *A. thaliana*, even for regions with relatively few markers. This behavior is a direct consequence of the sample size: the COPDGene study dataset contains $7,993$ individuals, whereas the number of samples available in the *A. thaliana* datasets is much smaller, oscillating between 84 and 95.

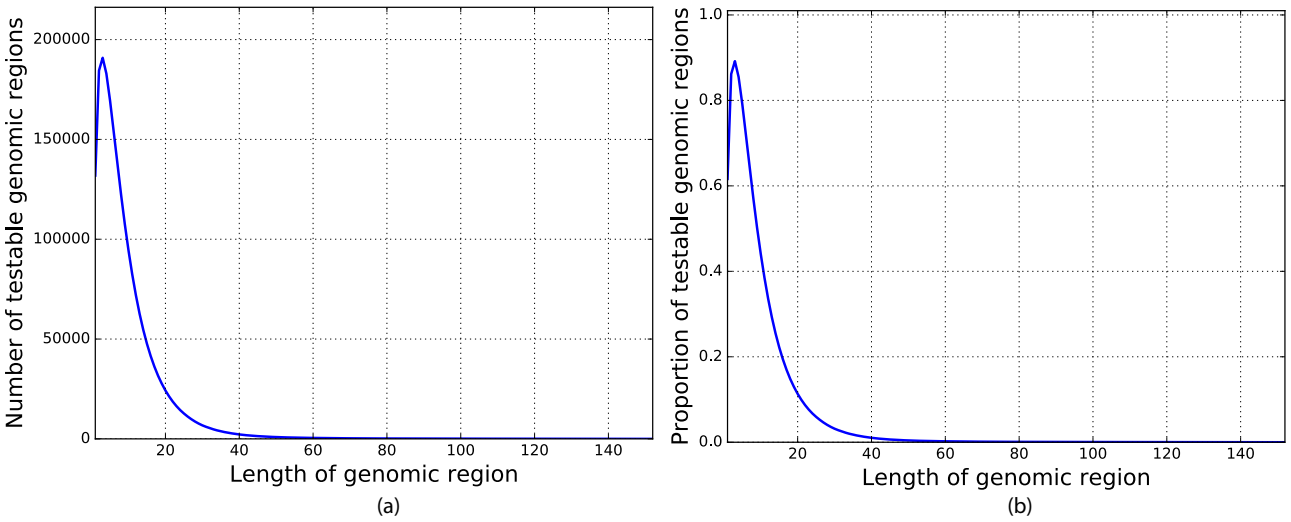

Figure S37: *A. thaliana*, *avrB* phenotype: (a) Total number of testable genomic regions. (b) Proportion of testable genomic regions.

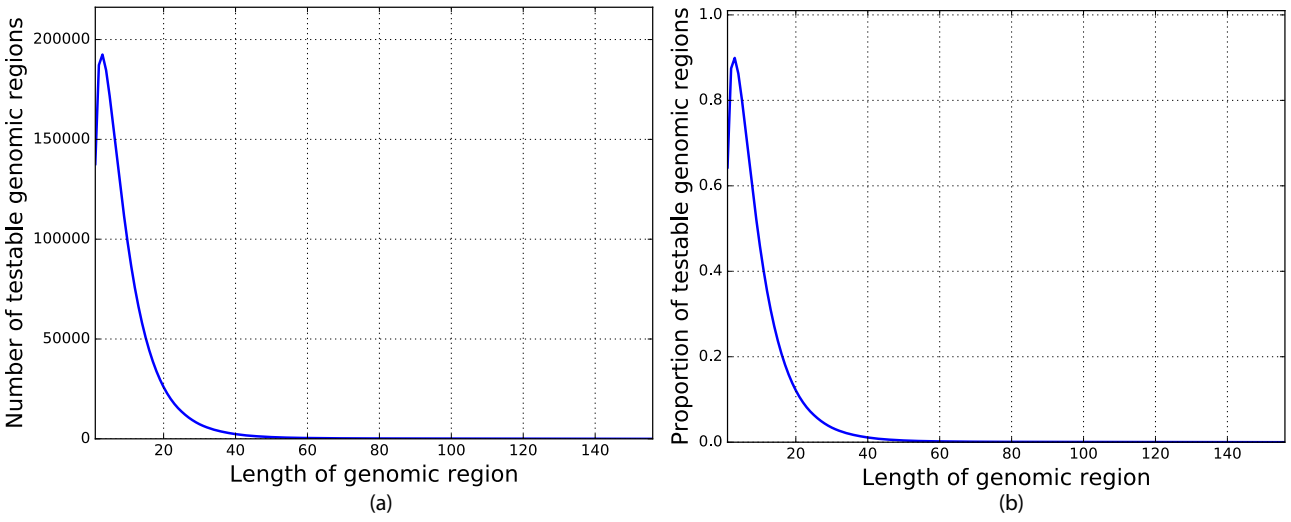

Figure S38: *A. thaliana*, *avrRpm1* phenotype: (a) Total number of testable genomic regions. (b) Proportion of testable genomic regions.

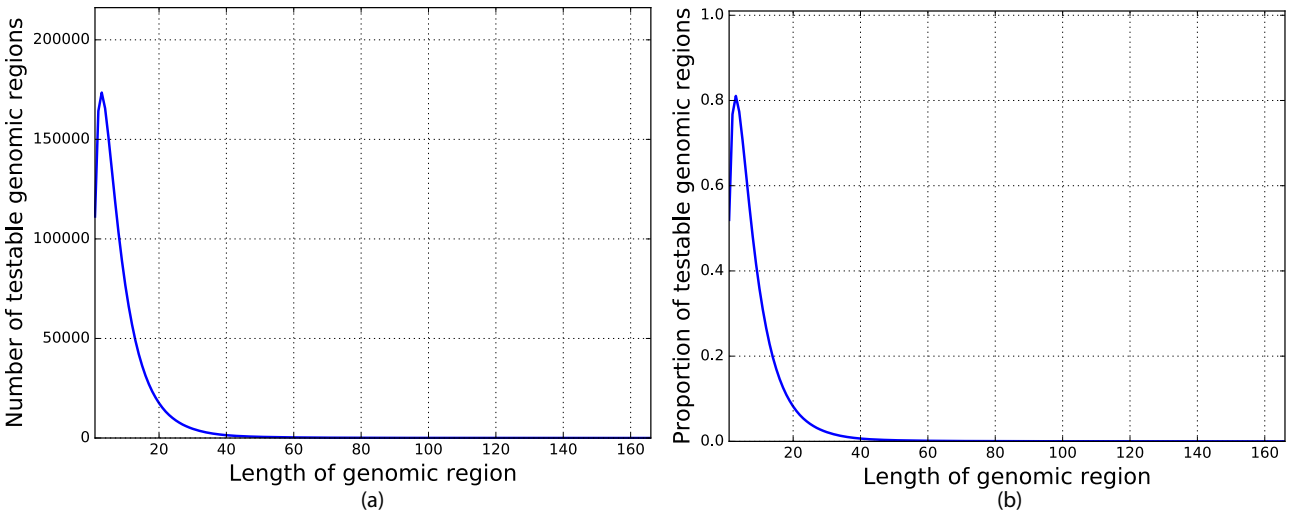

Figure S39: *A. thaliana*, *avrPphB* phenotype: (a) Total number of testable genomic regions. (b) Proportion of testable genomic regions.

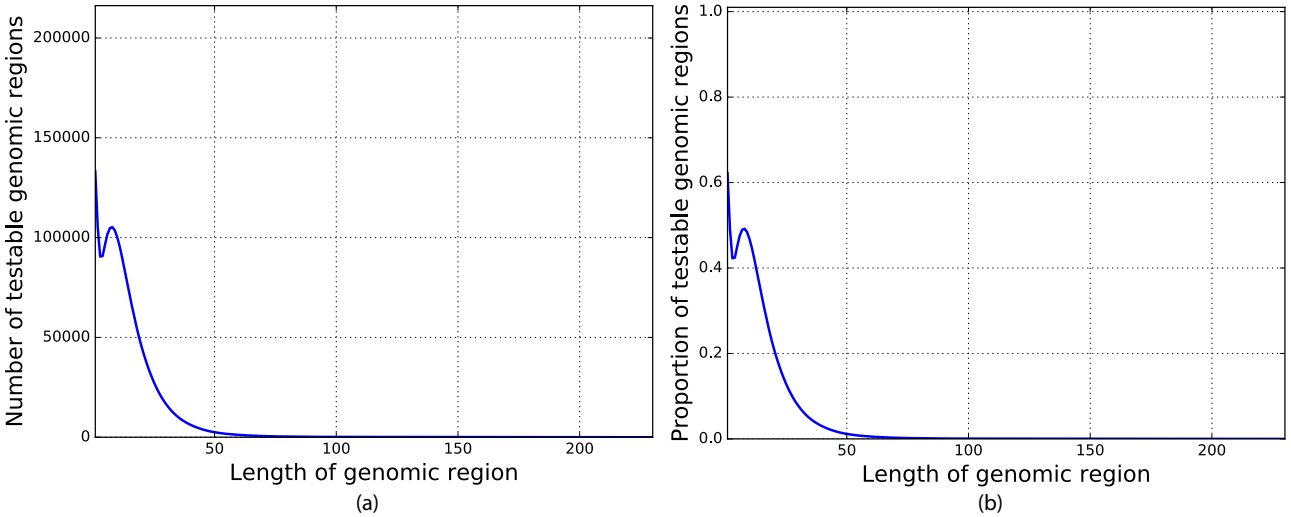

Figure S40: *A. thaliana*, LES phenotype: (a) Total number of testable genomic regions. (b) Proportion of testable genomic regions.

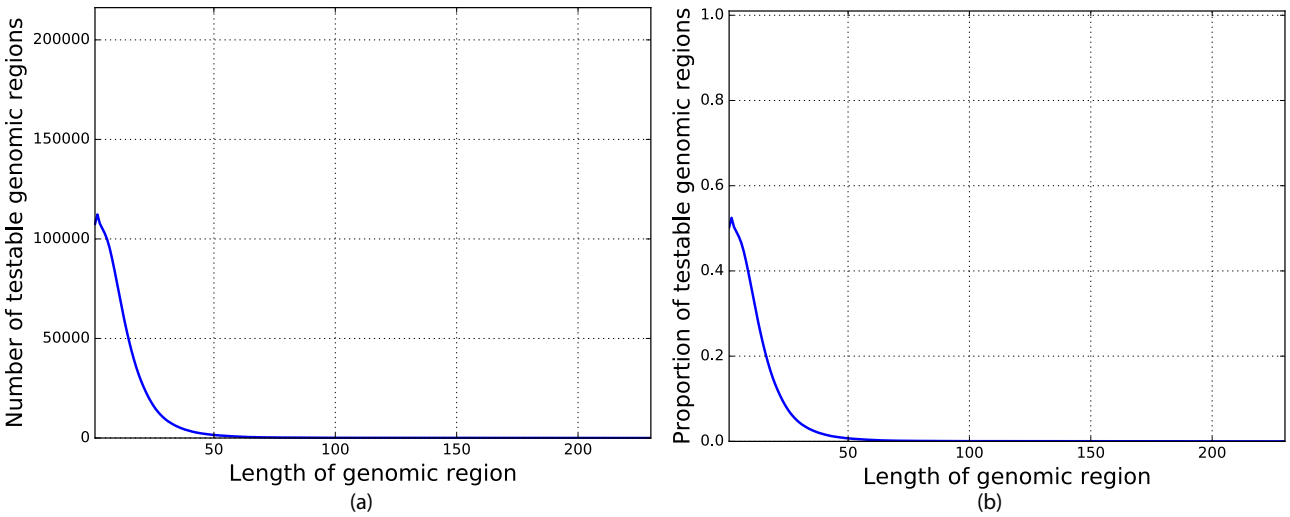

Figure S41: *A. thaliana*, LY phenotype: (a) Total number of testable genomic regions. (b) Proportion of testable genomic regions.

S4 The R package `fastcmh`

An R package `fastcmh` implementing the `FastCMH` algorithm is available on The Comprehensive R Archive Network at `https://CRAN.R-project.org/package=fastcmh`. The core of the algorithm is written in C++. The FDR extension described in Section S1.4 is implemented in the package.

S4.1 Installation in R

To install the R package `fastcmh`, open an R session and use the following command:

```
> install.packages("fastcmh", dependencies=TRUE)
```

This installs `fastcmh` along with its two main dependencies (`Rcpp` and `bindata`). In fact, a few other packages will also be installed, if they are not installed already (`R6`, `Rcpp`, `bindata`, `crayon`, `digest`, `e1071`, `fastcmh`, `magrittr`, `mvtnorm`, `praise`, `testthat`).

Note that the line above installs the packages to the default library folder for R; the location of this folder depends on the system and user settings.

The whole installation, depending on the internet speed, should take under one minute. Since the package is on CRAN, the installation will work for Windows, Mac and Linux.

S4.2 A short demo

After installation, in order to quickly see an example of `fastcmh` in action, there is a short demo which can be called via the commands:

```
> library(fastcmh)
> demofastcmh()
```

In order to see `FastCMH` in action, the user is prompted to press the `Enter` key to move onto the next step(s).

S4.3 Running `FastCMH`

After installing `fastcmh` (see Section S4.1), in an R session load the package using

```
> library("fastcmh")
```

Then, assuming that the data, label and covariate files are in the folder `/path/to/mydata/`, and the files are named `data.txt`, `label.txt` and `cov.txt`, respectively, then the command would be:

```
> mylist <- runfastcmh("/path/to/mydata", alpha=0.05)
```

where the significance threshold $\alpha = 0.05$ is used. There are many additional parameters that can be set, e.g. `doFDR=TRUE`, but the reader is referred to the documentation (see Section S4.4) for more information.

If the default file names have not been used — e.g. `mydata.txt`, `mylabel.txt` and `mycov.txt`, respectively, have been used — then this is no problem; simply specify the file names as in:

```
> mylist <- runfastcmh("/path/to/mydata", data="mydata.txt",
                       label="mylabel.txt", cov="mycov.txt",
                       alpha=0.05)
```

The output is the list `mylist`, and `mylist$sig` is a dataframe of the significant intervals (after filtering) with `start` indicating the starting point of the intervals, `end` the endpoint of the intervals and `pvalue` the p-value of the interval. See the package documentation for more details on the format of the output `mylist`.

S4.4 Documentation for `fastcmh`

While inside an R session, for quick access to all the information on any method (e.g. `runfastcmh`), check the `fastcmh` documentation with the command

```
> ?runfastcmh
```

in order to see more information on all the input/output parameters.

S5 Acknowledgements

COPDGene Investigators – Core Units

Administrative Core: James Crapo, MD (PI), Edwin Silverman, MD, PhD (PI), Barry Make, MD, Elizabeth Regan, MD, PhD

Genetic Analysis Core: Terri Beaty, PhD, Nan Laird, PhD, Christoph Lange, PhD, Michael Cho, MD, Stephanie Santorico, PhD, John Hokanson, MPH, PhD, Dawn DeMeo, MD, MPH, Nadia Hansel, MD, MPH, Craig Hersh, MD, MPH, Peter Castaldi, MD, MSc, Merry-Lynn McDonald, PhD, Emily Wan, MD, Megan Hardin, MD, Jacqueline Hetmanski, MS, Margaret Parker, MS, Marilyn Foreman, MD, Brian Hobbs, MD, Robert Busch, MD, Adel El-Boueiz, MD, Peter Castaldi, MD, Megan Hardin, MD, Dandi Qiao, PhD, Elizabeth Regan, MD, Eitan Halper-Stromberg, Ferdouse Begum, Sungho Won, Sharon Lutz, PhD

Imaging Core: David A Lynch, MB, Harvey O Coxson, PhD, MeiLan K Han, MD, MS, MD, Eric A Hoffman, PhD, Stephen Humphries MS, Francine L Jacobson, MD, Philip F Judy, PhD, Ella A Kazerooni, MD, John D Newell, Jr., MD, Elizabeth Regan, MD, James C Ross, PhD, Raul San Jose Estepar, PhD, Berend C Stoel, PhD, Juerg Tschirren, PhD, Eva van Rikxoort, PhD, Bram van Ginneken, PhD, George Washko, MD, Carla G Wilson, MS, Mustafa Al Qaisi, MD, Teresa Gray, Alex Kluiber, Tanya Mann, Jered Sieren, Douglas Stinson, Joyce Schroeder, MD, Edwin Van Beek, MD, PhD

PFT QA Core, Salt Lake City, UT: Robert Jensen, PhD

Data Coordinating Center and Biostatistics, National Jewish Health, Denver, CO: Douglas Everett, PhD, Anna Faino, MS, Matt Strand, PhD, Carla Wilson, MS

Epidemiology Core, University of Colorado Anschutz Medical Campus, Aurora, CO: John E. Hokanson, MPH, PhD, Gregory Kinney, MPH, PhD, Sharon Lutz, PhD, Kendra Young PhD, Katherine Pratte, MSPH, Lindsey Duca, MS

COPDGene Investigators – Clinical Centers

Ann Arbor VA: Jeffrey L. Curtis, MD, Carlos H. Martinez, MD, MPH, Perry G. Pernicano, MD

Baylor College of Medicine, Houston, TX: Nicola Hanania, MD, MS, Philip Alapat, MD, Venkata Bandi, MD, Mustafa Atik, MD, Aladin Boriek, PhD, Kalpatha Guntupalli, MD, Elizabeth Guy, MD, Amit Parulekar, MD, Arun Nachiappan, MD

Brigham and Women's Hospital, Boston, MA: Dawn DeMeo, MD, MPH, Craig Hersh, MD, MPH, George Washko, MD, Francine Jacobson, MD, MPH

Columbia University, New York, NY: R. Graham Barr, MD, DrPH, Byron Thomashow, MD, John Austin, MD, Belinda D'Souza, MD, Gregory D.N. Pearson, MD, Anna Rozenshtein, MD, MPH, FACR

Duke University Medical Center, Durham, NC: Neil MacIntyre, Jr., MD, Lacey Washington, MD, H. Page McAdams, MD

Health Partners Research Foundation, Minneapolis, MN: Charlene McEvoy, MD, MPH, Joseph Tashjian, MD

Johns Hopkins University, Baltimore, MD: Robert Wise, MD, Nadia Hansel, MD, MPH, Robert Brown, MD, Karen Horton, MD, Nirupama Putcha, MD, MHS,

Los Angeles Biomedical Research Institute at Harbor UCLA Medical Center, Torrance, CA: Richard Casaburi, PhD, MD, Alessandra Adami, PhD, Janos Porszasz, MD, PhD, Hans Fischer, MD, PhD, Matthew Budoff, MD, Harry Rossiter, PhD

Michael E. DeBakey VAMC, Houston, TX: Amir Sharafkhaneh, MD, PhD, Charlie Lan, DO

Minneapolis VA: Christine Wendt, MD, Brian Bell, MD

Morehouse School of Medicine, Atlanta, GA: Marilyn Foreman, MD, MS, Gloria Westney, MD, MS, Eugene Berkowitz, MD, PhD

National Jewish Health, Denver, CO: Russell Bowler, MD, PhD, David Lynch, MD

Reliant Medical Group, Worcester, MA: Richard Rosiello, MD, David Pace, MD

Temple University, Philadelphia, PA: Gerard Criner, MD, David Ciccolella, MD, Francis Cordova, MD, Chandra Dass, MD, Gilbert D'Alonzo, DO, Parag Desai, MD, Michael Jacobs, PharmD, Steven Kelsen, MD, PhD, Victor Kim, MD, A. James Mamary, MD, Nathaniel Marchetti, DO, Aditi Satti, MD, Kartik Shenoy, MD, Robert M. Steiner, MD, Alex Swift, MD, Irene Swift, MD, Maria Elena Vega-Sanchez, MD

University of Alabama, Birmingham, AL: Mark Dransfield, MD, William Bailey, MD, J. Michael Wells, MD, Surya Bhatt, MD, Hrudaya Nath, MD

University of California, San Diego, CA: Joe Ramsdell, MD, Paul Friedman, MD, Xavier Soler, MD, PhD, Andrew Yen, MD

University of Iowa, Iowa City, IA: Alejandro Cornellas, MD, John Newell, Jr., MD, Brad Thompson, MD

University of Michigan, Ann Arbor, MI: MeiLan Han, MD, Ella Kazerooni, MD, Carlos Martinez, MD

University of Minnesota, Minneapolis, MN: Joanne Billings, MD, Tadashi Allen, MD

University of Pittsburgh, Pittsburgh, PA: Frank Sciurba, MD, Divay Chandra, MD, MSc, Joel Weissfeld, MD, MPH, Carl Fuhrman, MD, Jessica Bon, MD

University of Texas Health Science Center at San Antonio, San Antonio, TX: Antonio Anzueto, MD, Sandra Adams, MD, Diego Maselli-Caceres, MD, Mario E. Ruiz, MD

References

Benjamini, Y. and Hochberg, Y. (1995a). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, **57**(1), 289–300.

Benjamini, Y. and Hochberg, Y. (1995b). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B*, **57**(1), 289–300.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, **29**(4), 1165–1188.

Casale, F. P., Rakitsch, B., Lippert, C., and Stegle, O. (2015). Efficient set tests for the genetic analysis of correlated traits. *Nature methods*, **12**(8), 755–758.

Cho, M. H., McDonald, M.-L. N., Zhou, X., Mattheisen, M., Castaldi, P. J., Hersh, C. P., DeMeo, D. L., Sylvia, J. S., Ziniti, J., Laird, N. M., Lange, C., Litonjua, A. A., Sparrow, D., Casaburi, R., Barr, R. G., Regan, E. A., Make, B. J., Hokanson, J. E., Lutz, S., Dudenkov, T. M., Farzadegan, H., Hetmanski, J. B., Tal-Singer, R., Lomas, D. A., Bakke, P., Gulsvik, A., Crapo, J. D., Silverman, E. K., and Beaty, T. H. (2014). Risk loci for chronic obstructive pulmonary disease: a genome-wide association study and meta-analysis. *The Lancet Respiratory Medicine*, **2**(3), 214–225.

Gilbert, P. B. (2005). A modified false discovery rate multiple-comparisons procedure for discrete data, applied to human immunodeficiency virus genetics. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **54**(1), 143–158.

Llinares-López, F., Grimm, D. G., Bodenham, D. A., Gieraths, U., Sugiyama, M., Rowan, B., and Borgwardt, K. (2015). Genome-wide detection of intervals of genetic heterogeneity associated with complex traits. *Bioinformatics*, **31**(12), i240–i249.

Marchini, J., Cardon, L. R., Phillips, M. S., and Donnelly, P. (2004). The effects of human population structure on large genetic association studies. *Nature genetics*, **36**(5), 512–517.

Papaxanthos, L., Llinares-López, F., Bodenham, D. A., and Borgwardt, K. (2016). Finding significant combinations of features in the presence of categorical covariates. In *Advances in Neural Information Processing Systems 29: 30th Annual Conference on Neural Information Processing Systems 2016. Print in process.*

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

Qiu, W., Cho, M. H., Riley, J. H., Anderson, W. H., Singh, D., Bakke, P., Gulsvik, A., Litonjua, A. A., Lomas, D. A., Crapo, J. D., *et al.* (2011). Genetics of sputum gene expression in chronic obstructive pulmonary disease. *PLoS One*, **6**(9), e24395.

Schmid, K. and Yang, Z. (2008). The trouble with sliding windows and the selective pressure in brca1. *PLoS One*, **3**(11), e3746.

Song, M., Hao, W., and Storey, J. D. (2015). Testing for genetic associations in arbitrarily structured populations. *Nature genetics*, **47**(5), 550–554.

Tarone, R. E. (1990). A Modified Bonferroni Method for Discrete Data. *Biometrics*, **46**(2), 515.