

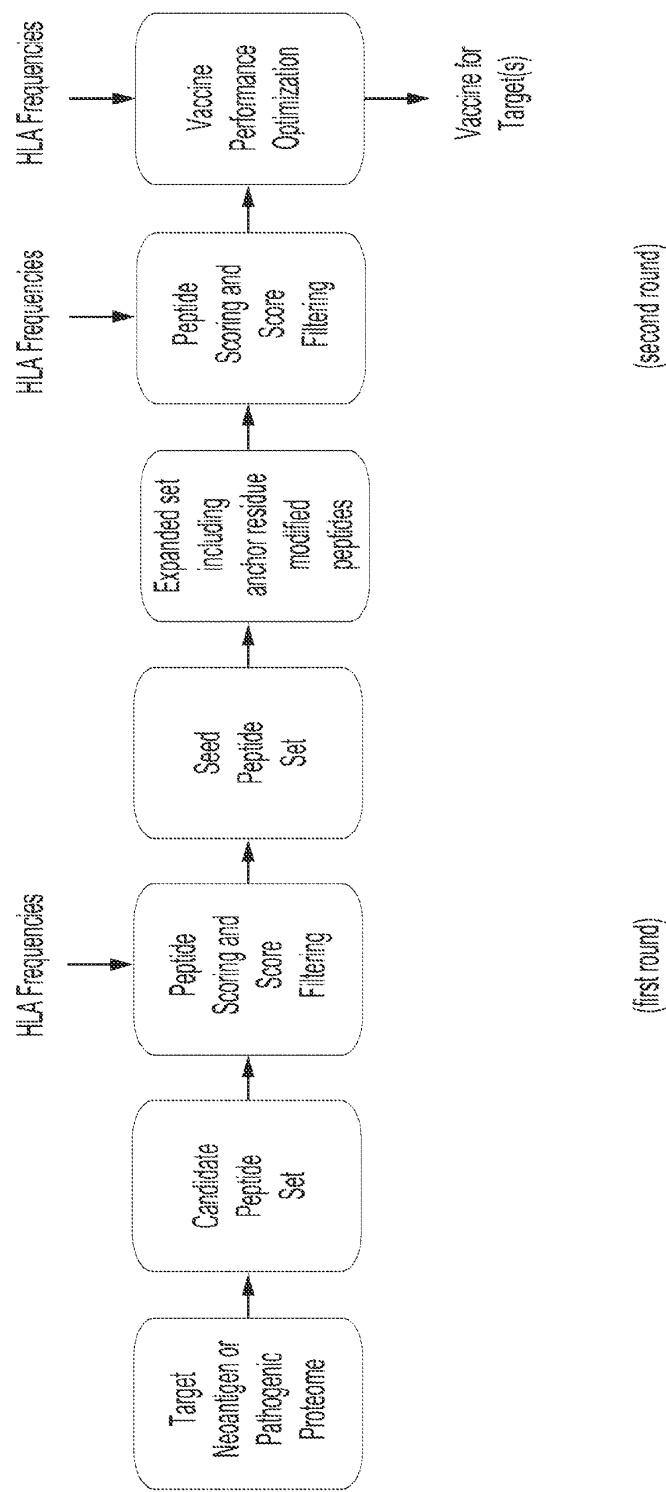


FIG. 1

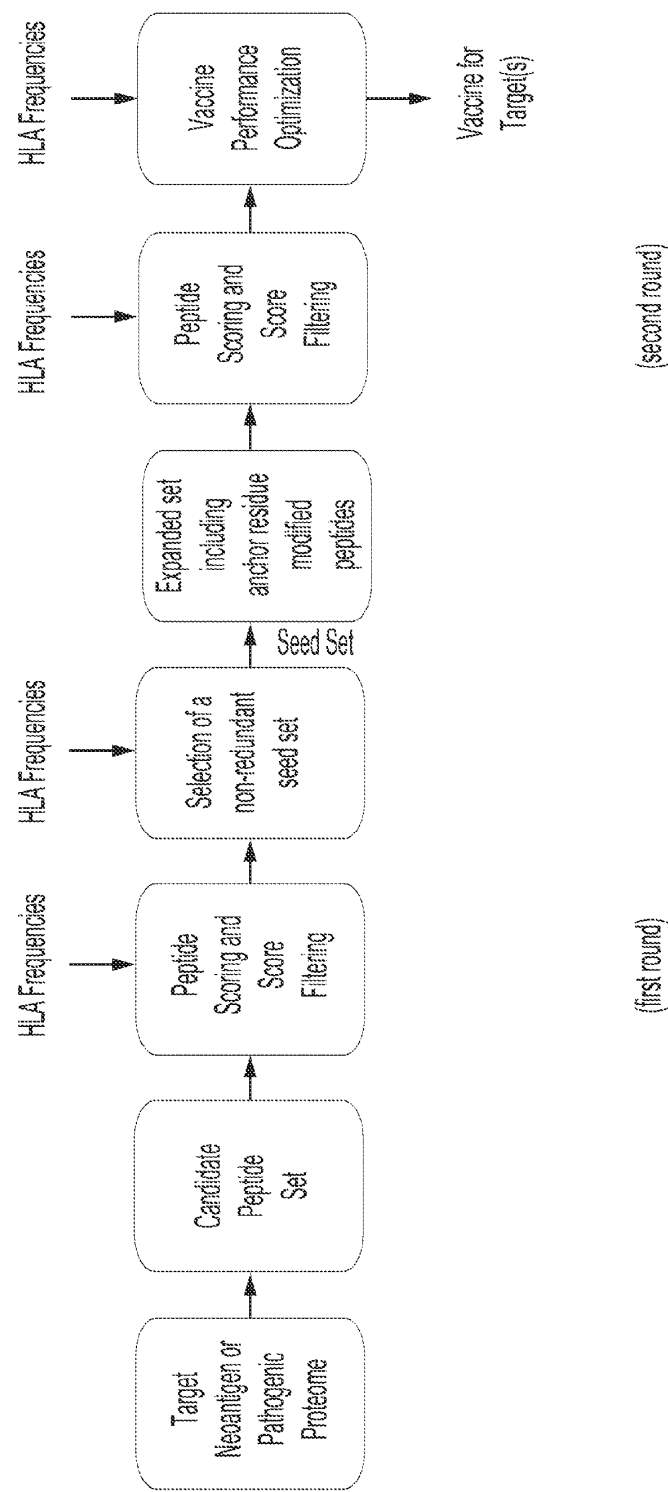


FIG. 2

3/14

Individual KRAS Vaccines G12D/G12V/G12R/G12C/G13D (MHC Class I)

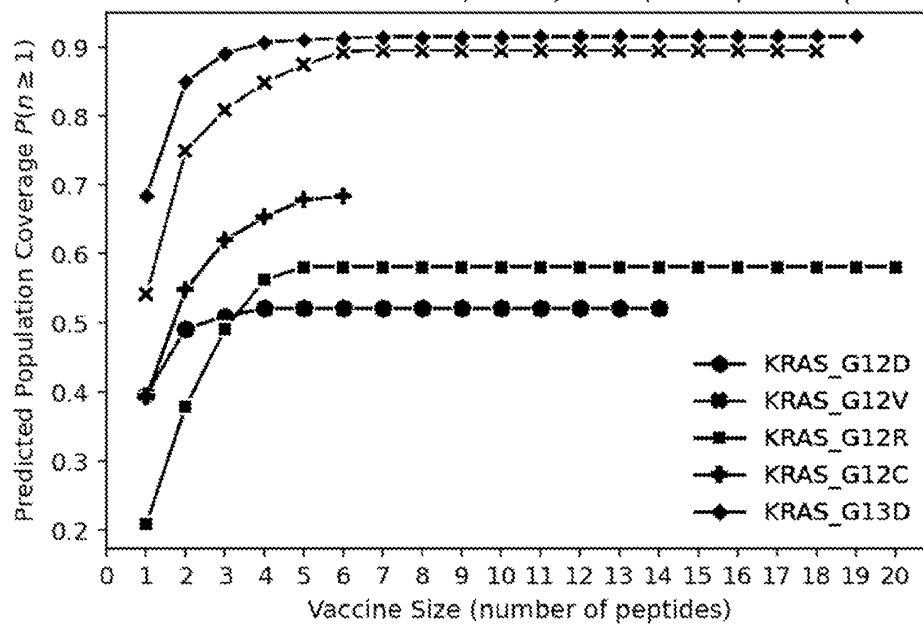


FIG. 3

4/14

Individual KRAS Vaccines G12D/G12V/G12R/G12C/G13D (MHC Class II)

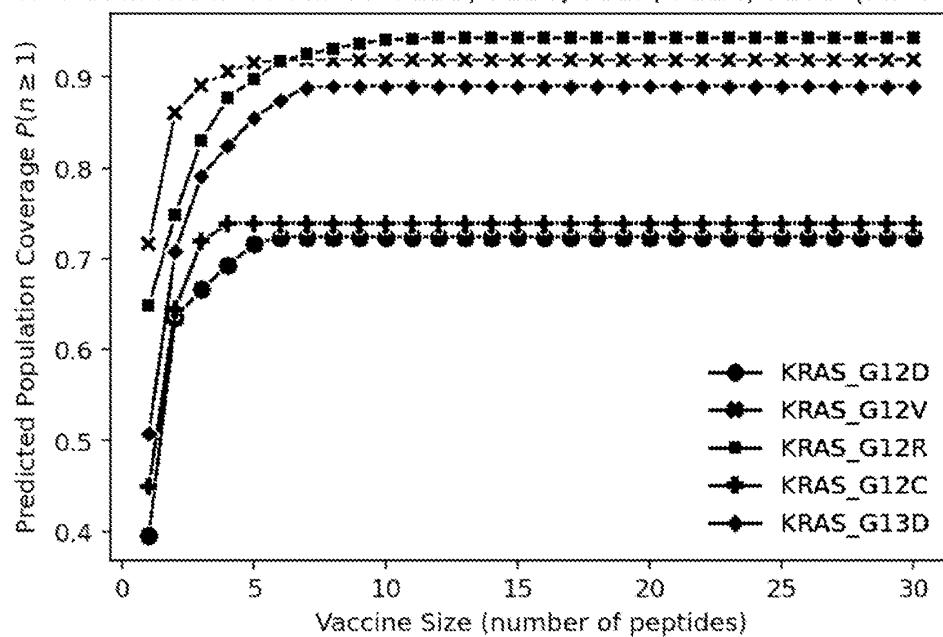


FIG. 4

5/14

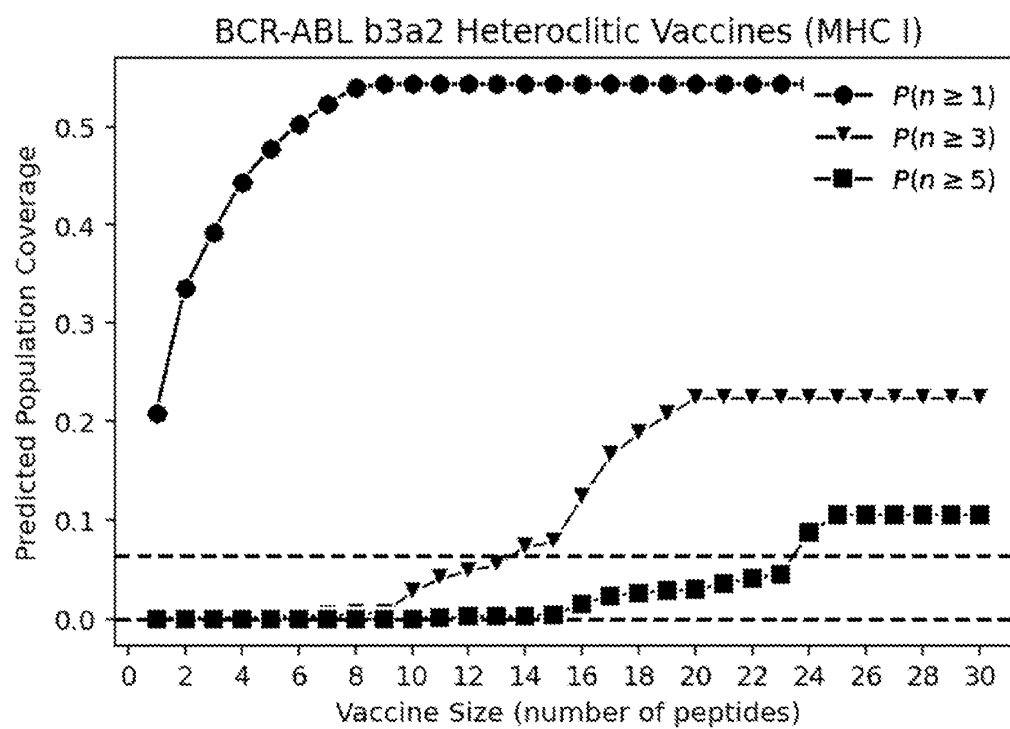


FIG. 5

6/14

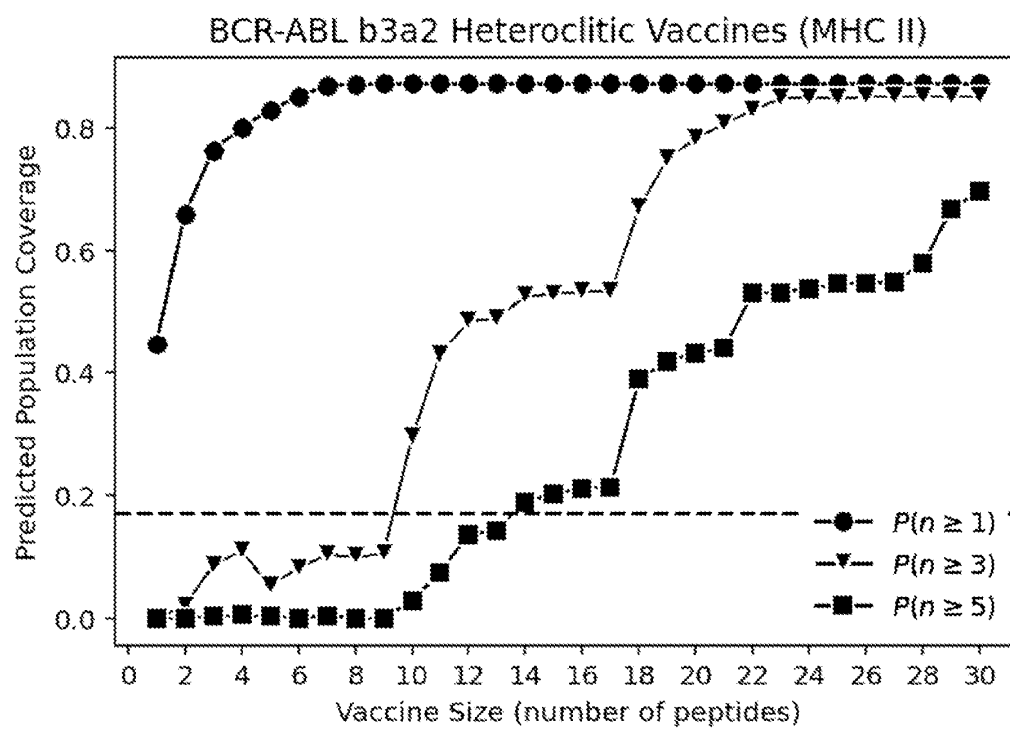


FIG. 6

7/14

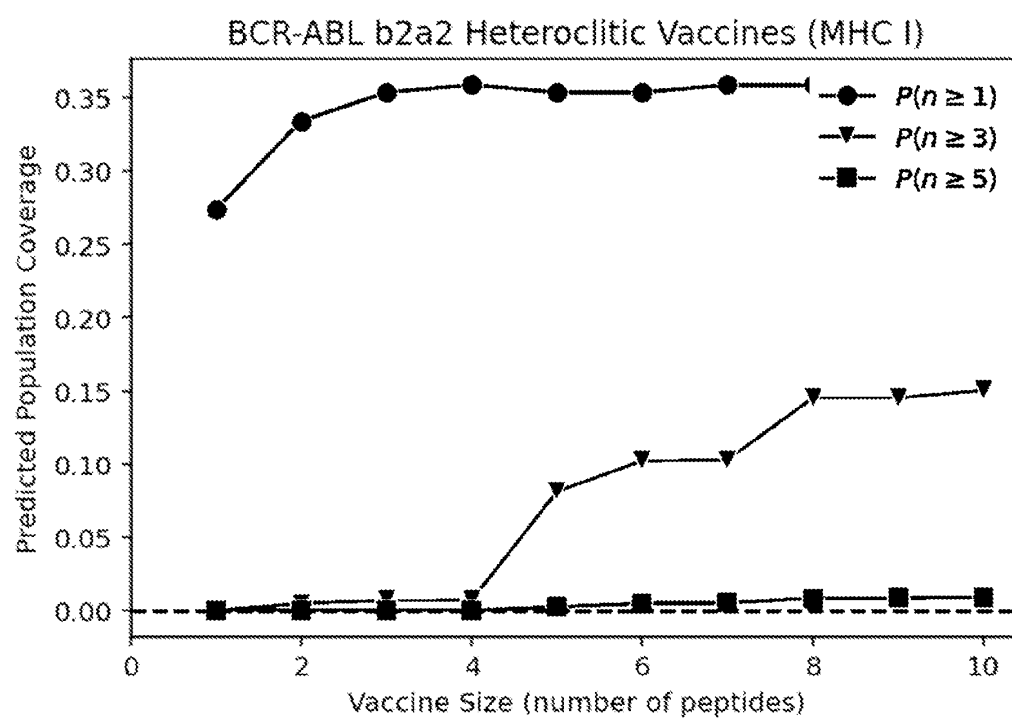


FIG. 7

8/14

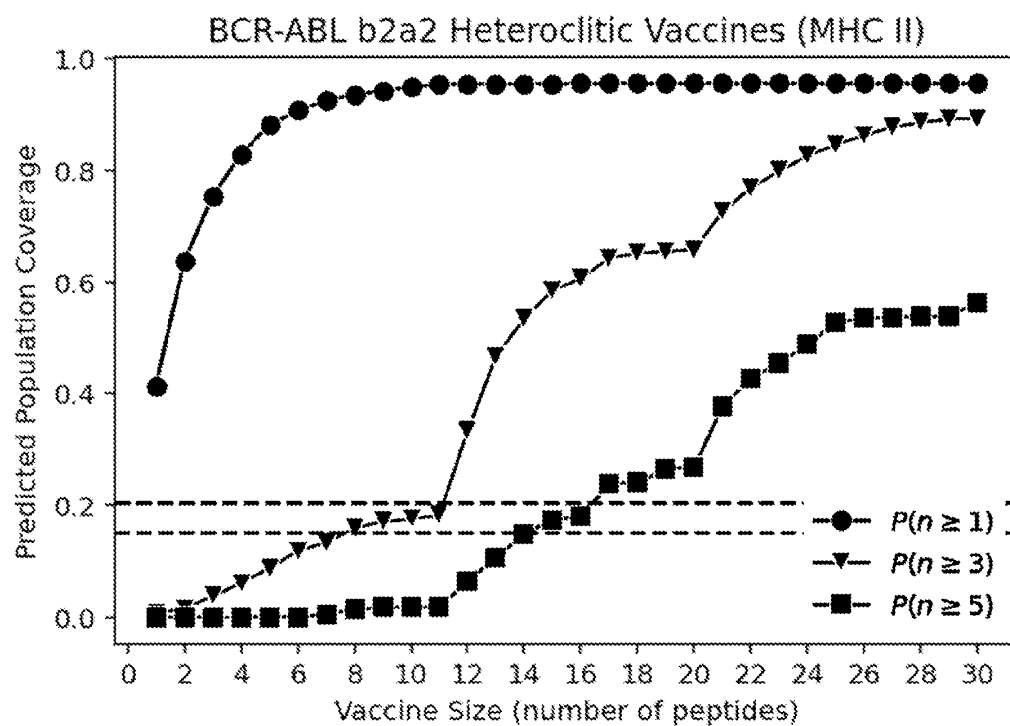


FIG. 8

9/14

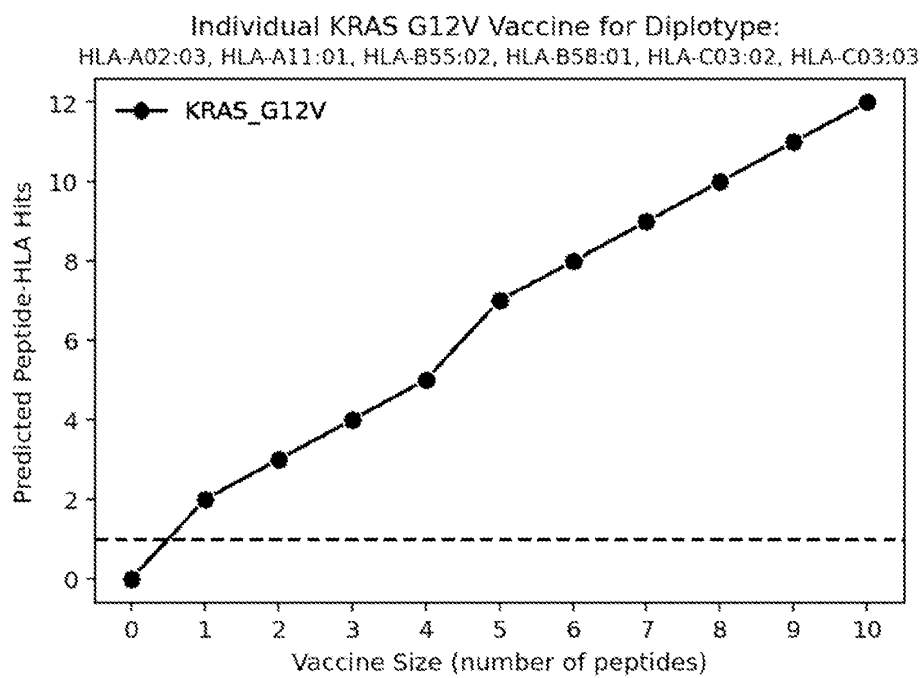


FIG. 9

10/14

Factoring of disease presentation type probabilities and for each presentation, probability of targets presented

	Brain Cancer	Bronchus and Lung Cancer	Colorectal Cancer	Pancreatic Cancer	Skin Cancer	Thyroid Cancer	Breast Cancer	Ovarian Cancer
BRAF_V600E	-	-	0.1029	-	0.4392	0.5854	-	-
BRAF_V600M	-	-	-	-	0.0853	-	-	-
EGFR_A289V	0.0230	-	-	-	-	-	-	-
EGFR_G598V	0.0208	-	-	-	-	-	-	-
EGFR_L858R	-	0.0216	-	-	-	-	-	-
HRAS_Q61K	-	-	-	-	-	0.0061	-	-
HRAS_Q61R	-	-	-	-	-	0.0285	-	-
IDH1_R132C	0.0197	-	-	-	-	-	-	-
IDH1_R132H	0.4178	-	-	-	-	-	-	-
KRAS_G12A	-	0.0169	-	-	-	-	-	-
KRAS_G12C	-	0.0583	-	-	-	-	-	-
KRAS_G12D	-	0.0188	0.1235	0.3280	-	-	-	-
KRAS_G12R	-	-	-	0.1505	-	-	-	-
KRAS_G12V	-	0.0376	0.0947	0.2258	-	-	-	0.0092
KRAS_G13D	-	-	0.0700	-	-	-	-	-
NRAS_Q61K	-	-	-	-	0.0832	0.0163	-	-
NRAS_Q61L	-	-	-	-	0.0384	-	-	-
NRAS_Q61R	-	-	-	-	0.1215	0.0630	-	0.0069
PIK3CA_E542K	-	0.0169	-	-	-	-	0.0435	-
PIK3CA_E545K	-	0.0197	0.0700	-	-	-	0.0638	-
PIK3CA_H1047R	-	-	-	-	-	-	0.1225	-
TP53_R158L	-	0.0188	-	-	-	-	-	-
TP53_R175H	0.0186	-	0.0679	-	-	-	0.0202	0.0343
TP53_R248Q	-	-	-	-	-	-	-	0.0252
TP53_R273C	0.0581	-	-	-	-	-	0.0051	0.0137
TP53_R273H	-	-	-	-	-	-	0.0121	0.0206

FIG. 10

SUBSTITUTE SHEET (RULE 26)

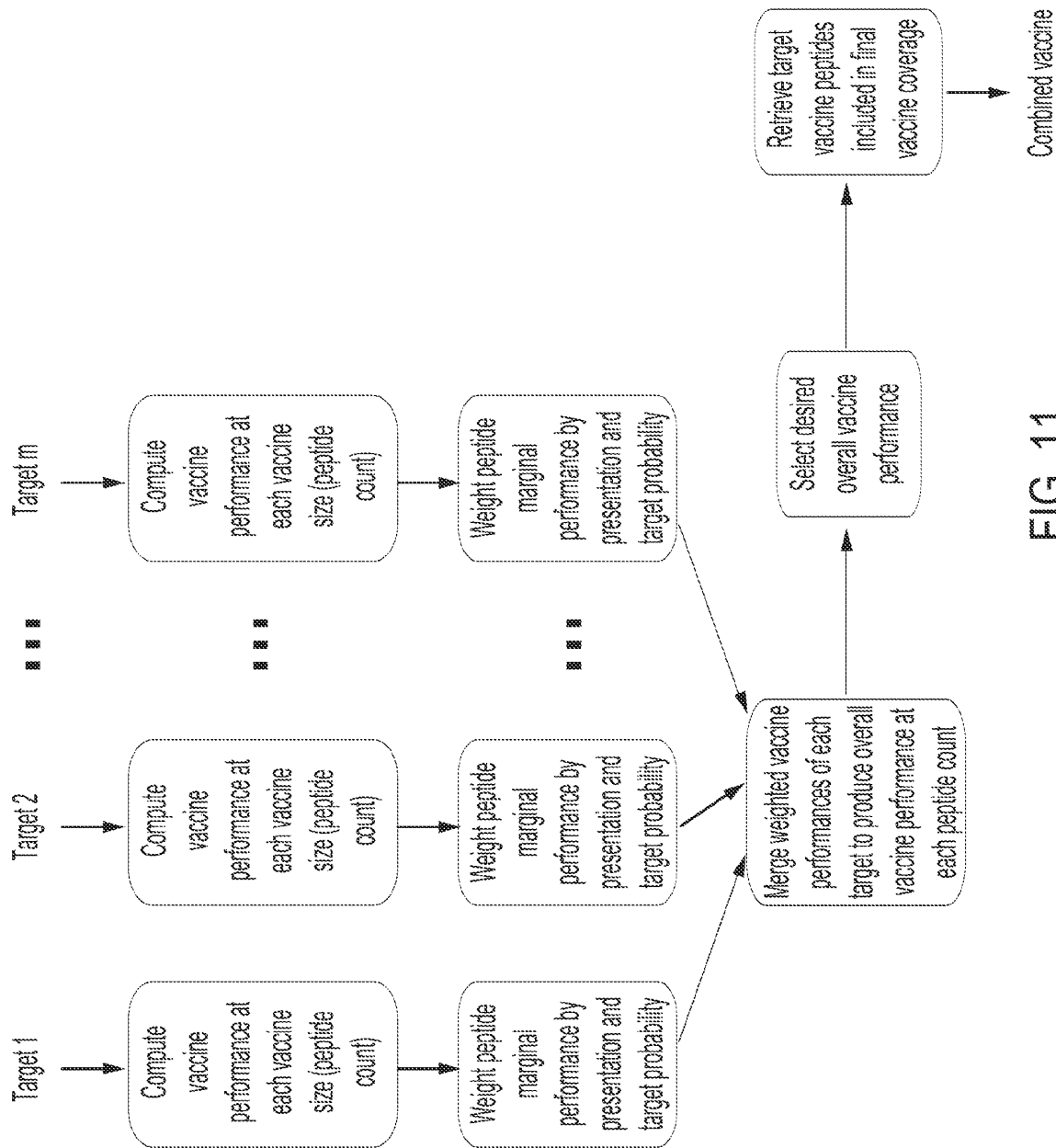


FIG. 11

12/14

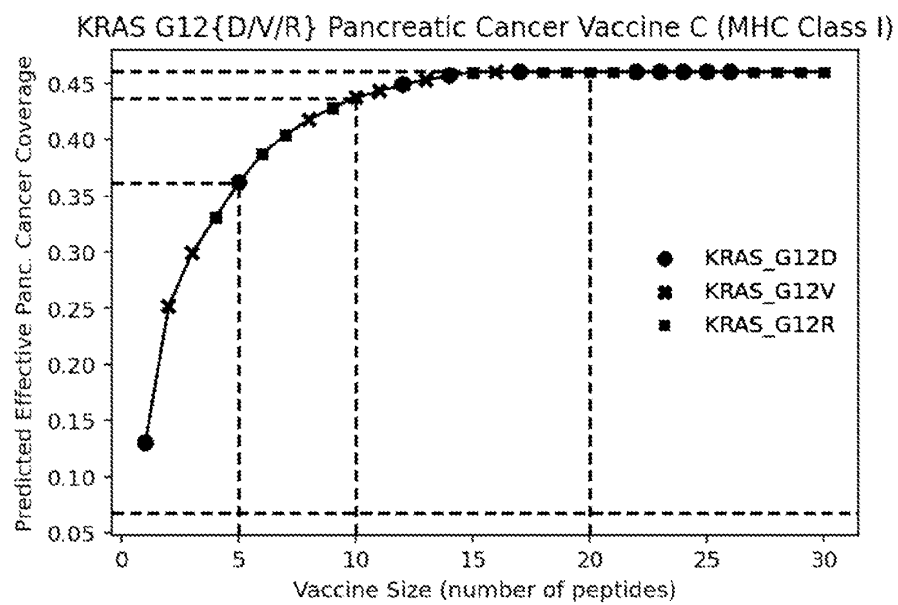


FIG. 12

13/14

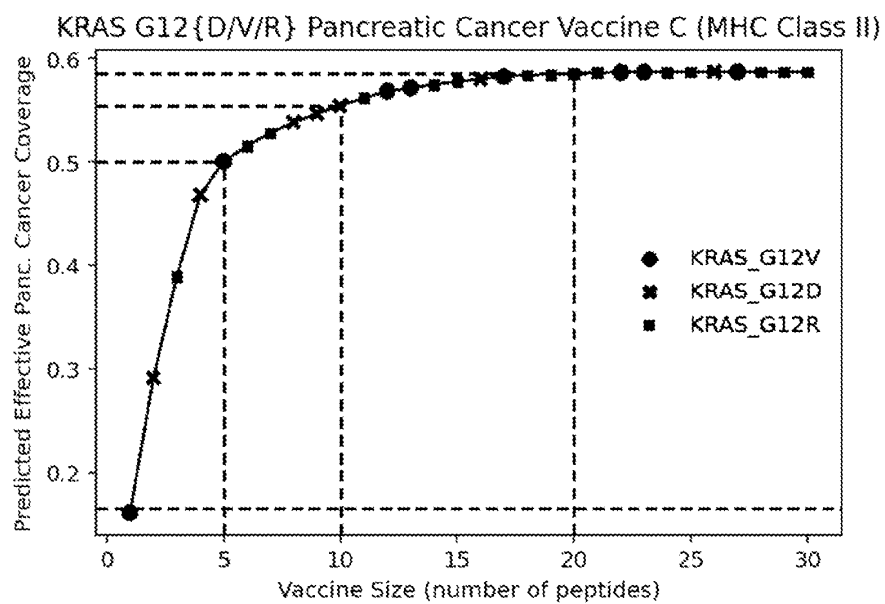


FIG. 13

14/14

```
def merge_multi(lists):
    values = []

    # While any list in lists has elements remaining
    while max(map(lambda l: len(l), lists)) > 0:
        # Find list with largest value at its head.
        cur_max = None
        cur_max_idx = None
        for idx, l in enumerate(lists):
            if not l: # List is empty.
                continue
            if cur_max is None or l[0] > cur_max:
                cur_max = l[0]
                cur_max_idx = idx
        # Pop that value from list l.
        values.append((lists[cur_max_idx].pop(0), cur_max_idx))

    return values
```

FIG. 14