In [1]:
```
import hail as hl

# hl.init(backend="spark")

# Say, "the default sans-serif font is COMIC SANS"
# matplotlib.rcParams['font.sans-serif'] = "Arial"
# Then, "ALWAYS use sans-serif fonts"
# matplotlib.rcParams['font.family'] = "sans-serif"
```

In [2]:
```
# import holoviews as hv
import numpy as np
import pandas as pd

from bokeh.layouts import column, gridplot, Plot, row
from bokeh.models import *
from bokeh.plotting import *
from bokeh.palettes import Category10, Spectral6
from gnomad import *
from gnomad.sample_qc.ancestry import POP_COLORS
from gnomad.utils.filtering import add_filters_expr
from gnomad.utils.plotting import *
# from gnomad_qc.v3.resources import *
from IPython.core.display import display, HTML
from statsmodels.robust.scale import mad

# hv.extension("bokeh")

TOOLS = "hover,save,pan,box_zoom,reset,wheel_zoom"
output_notebook()
display(HTML("<style>.container { width:100% !important; }</style>"))

if 'old_show' not in dir():
    old_show = hl.Table.show
    def new_show(t, n=10, width=90, truncate=None, types=True):
        old_show(t, 10, 170, 40)
    hl.Table.show = new_show
```

```
/tmp/ipykernel_50791/3819452698.py:14: DeprecationWarning: Importing display
from IPython.core.display is deprecated since IPython 7.14, please import fro
m IPython display
  from IPython.core.display import display, HTML
```

BokehJS 3.1.1 successfully loaded.

In [ ]:
```
dragen_called_rgp = hl.read_matrix_table('gs://marten-seqr-sandbox-storage/c
```

In [ ]:
```
gatk_called_rgp = hl.read_matrix_table('gs://marten-seqr-sandbox-storage/gat
```

In [ ]:
```
# dragen_massive_mt = hl.import_vcf('gs://seqr-scratch-temp/gregor_consortiu
#                                   reference_genome="GRCh38",
#                                   force_bgz=True,
#                                   array_elements_required=False)
```

In [ ]:
```
# dragen_massive_mt = dragen_massive_mt.repartition(1500)
```

In [3]:
```
dragen_massive_mt = hl.read_matrix_table('gs://seqr-scratch-temp/gregor_cons
```

```
Initializing Hail with default parameters...
/opt/conda/miniconda3/lib/python3.10/site-packages/hailtop/aiocloud/aiogoogl
e/user_config.py:43: UserWarning: Reading spark-defaults.conf to determine GC
S requester pays configuration. This is deprecated. Please use `hailctl confi
g set gcs_requester_pays/project` and `hailctl config set gcs_requester_pays/
buckets`.
  warnings.warn(
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogL
evel(newLevel).
SPARKMONITOR_LISTENER: Started SparkListener for Jupyter Notebook
SPARKMONITOR_LISTENER: Port obtained from environment: 41569
SPARKMONITOR_LISTENER: Application Started: application_1712844629510_0003
...Start Time: 1712864136105
Running on Apache Spark version 3.3.0
SparkUI available at http://dmdragen-m.us-central1-b.c.marten-seqr-sandbox-91
23.internal:34271
Welcome to
     __  __     <>__
    / /_/ /__  __/ /
   / __  / _ `/ / /
  /_/ /_/\_,_/_/_/   version 0.2.120-f00f916faf78
LOGGING: writing to /home/hail/hail-20240411-1935-0.2.120-f00f916faf78.log
```

In [4]:
```
dragen_massive_mt.describe()
```

```
----------------------------------------
Global fields:
    None
----------------------------------------
Column fields:
    's': str
----------------------------------------
Row fields:
    'locus': locus<GRCh38>
    'alleles': array<str>
    'rsid': str
    'qual': float64
    'filters': set<str>
    'info': struct {
        AC: array<int32>,
        AF: array<float64>,
        AN: int32,
        AS_QUALapprox: str,
        AS_YNG: array<str>,
        CALIBRATION_SENSITIVITY: array<str>,
        QUALapprox: int32,
        SCORE: array<str>
    }
----------------------------------------
Entry fields:
    'AD': array<int32>
    'FT': str
    'GQ': int32
    'GT': call
    'RGQ': int32
----------------------------------------
Column key: ['s']
Row key: ['locus', 'alleles']
----------------------------------------
```

In [ ]:
```
# Check Coverage
dragen_massive_mt = dragen_massive_mt.annotate_entries(
    total_cov = hl.if_else(~hl.is_missing(dragen_massive_mt.AD),dragen_massi
)
```

In [ ]:
```
total_cov_dragen_obj = hl.plot.histogram(dragen_massive_mt.total_cov,range=(
```

In [ ]:
```
show(total_cov_dragen_obj)
```

In [ ]:
```
dragen_massive_mt.total_cov.summarize()
```

In [ ]:
```
# Check GQ Distribution
```

In [ ]:
```
show(hl.plot.histogram(dragen_massive_mt.GQ))
```

In [ ]:
```
dragen_massive_mt.GQ.summarize()
```

In [ ]:
```
# Check ploidy in sex chromosomes
```

In [ ]:
```
dragen_massive_mt_autosomes = dragen_massive_mt.filter_rows((dragen_massive_
dragen_massive_mt_xy = dragen_massive_mt.filter_rows((dragen_massive_mt.locu

# mt = mt.filter_rows(mt.locus.contig != 'Y')
```

In [ ]:
```
dragen_massive_mt_xy = dragen_massive_mt_xy.annotate_entries(
    diploidy = dragen_massive_mt_xy.GT.is_diploid()
)
```

In [ ]:
```
dragen_massive_mt_xy.aggregate_entries(hl.agg.counter(dragen_massive_mt_xy.c
```

In [ ]:
```
dragen_massive_mt_xy.entries().head(100).show(100)
```

In [ ]:
```
dragen_massive_mt_xy.entries().tail(100).show(100)
```

In [ ]:
```
dragen_massive_mt_xy.aggregate_rows(hl.agg.counter(dragen_massive_mt_xy.locu
```

In [ ]:
```
# Check Filters and FT fields
```

In [ ]:
```
dragen_massive_mt.aggregate_rows(hl.agg.counter(dragen_massive_mt.filters))
```

In [ ]:
```
dragen_massive_mt.aggregate_entries(hl.agg.counter(dragen_massive_mt.FT))
```

In [ ]:
```
# Per sample and silliest sample qc (the easy way) work
```

In [ ]:
```
dragen_massive_mt.aggregate_rows(hl.agg.counter(dragen_massive_mt.locus.cont
```

In [ ]:
```
# dragen_massive_mt_entries = dragen_massive_mt.entries()
```

In [ ]:
```
# dragen_massive_mt_entries.aggregate(hl.agg.counter(dragen_massive_mt_entri
```

In [ ]:
```
dragen_massive_sqc_cols = hl.sample_qc(dragen_massive_mt).cols()
```

In [ ]:
```
dragen_massive_sqc_cols.show()
```

In [ ]:
```
dragen_massive_sqc_cols.sample_qc.n_called.summarize()
```

In [ ]:
```
dragen_massive_sqc_cols.sample_qc.n_non_ref.summarize()
```

In [ ]:
```
dragen_massive_sqc_cols.sample_qc.n_singleton.summarize()
```

In [ ]:
```
# Odd question - so all calls are diploid, but is anything odd on sex chromo
```

In [ ]:
```
# dragen_massive_mt_xy.aggregate_rows(hl.agg.counter(dragen_massive_mt_xy.is
```

In [ ]:
```
dragen_massive_mt_xy.aggregate_rows(hl.agg.counter(dragen_massive_mt_xy.locu
```

In [ ]:
```
dragen_massive_mt_xy.aggregate_rows(hl.agg.counter(dragen_massive_mt_xy.locu
```

In [ ]:
```
dragen_massive_mt_xonly = dragen_massive_mt_xy.filter_rows(dragen_massive_mt

dragen_massive_mt_yonly = dragen_massive_mt_xy.filter_rows(dragen_massive_mt
```

In [ ]:
```
xonly_sqc = hl.sample_qc(dragen_massive_mt_xonly).cols()
```

In [ ]:
```
yonly_sqc = hl.sample_qc(dragen_massive_mt_yonly).cols()
```

In [ ]:
```
xonly_sqc.sample_qc.n_called.show()
```

In [ ]:
```
xonly_sqc.sample_qc.n_non_ref.summarize()
```

In [ ]:
```
yonly_sqc.sample_qc.n_non_ref.summarize()
```

In [ ]:
```
# Lets see what this callset has for dragen autosomal nonref
```

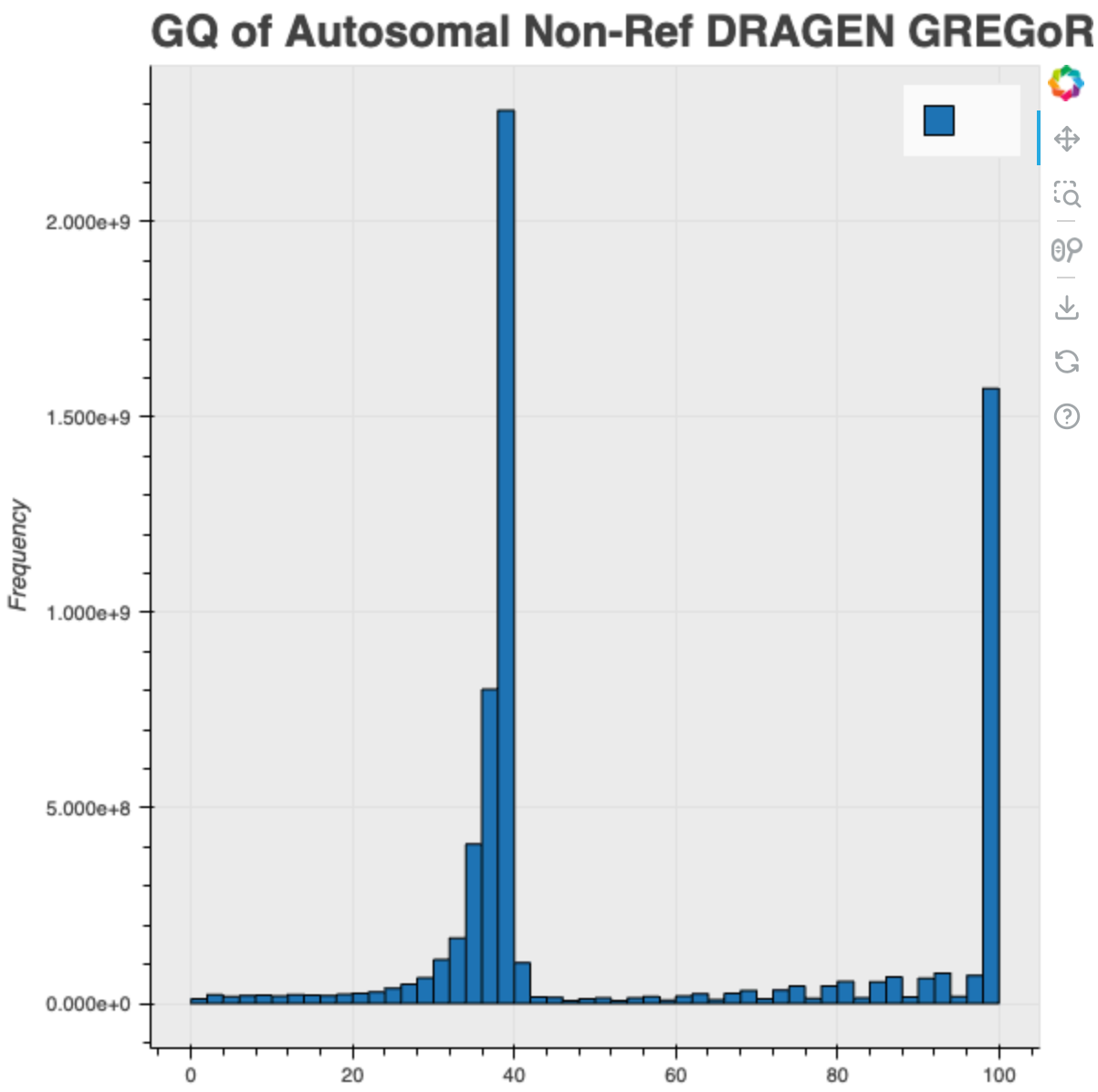
In [5]:
```
sex_set = hl.set({"chrX","chrY"})

dragen_massive_mt_autosomal_nonref = dragen_massive_mt.filter_rows(~sex_set.
```

In [6]:
```
dragen_massive_mt_autosomal_nonref = dragen_massive_mt_autosomal_nonref.filt
```

In [7]:
```
plotting_obj = hl.plot.histogram(dragen_massive_mt_autosomal_nonref.GQ,range
```

```
[Stage 1:=================================================>       (26 + 8) /
30]
```

In [8]:
```
plotting_obj.title = f"GQ of Autosomal Non-Ref DRAGEN GREGoR Anvil Request"
plotting_obj.title.text_font_size = '18pt'
show(plotting_obj)
```


GQ of Autosomal Non-Ref DRAGEN GREGoR
Frequency
2.000e+9
1.500e+9
1.000e+9
5.000e+8
0.000e+0
0
20
40
60
80
100

In [ ]: