Analyzing Clinical and Genomic Oncological Data with {genieBPC} and {gnomeR}

GENIE BPC Winter Symposium
January 16, 2024
Sammi Brown
Hannah Fuchs
Jessica Lavery
Karissa Whiting

Agenda

Clinico-Genomic Data Processing Pipeline

Case study

Clinical data processing with {genieBPC}

Genomic data processing with {gnomeR}

Conclusion

{genieBPC} & {gnomeR} R Packages

The {genieBPC} package is a pipeline to programmatically access the data corresponding to each release from Synapse to support reproducibility, and to create datasets linking clinical and genomic data for analysis.

Created and developed by

Samantha Brown

Michael Curry

Hannah Fuchs

Jessica Lavery

Axel Martin

Dan Sjoberg

Karissa Whiting

The {gnomeR} package provides a consistent framework for genetic data wrangling, processing, visualization and analysis.

Created and developed by

Arshi Arora

Michael Curry

Hannah Fuchs

Axel Martin

Karissa Whiting

Installation Instructions

Installing {genieBPC}:

```
install.packages("genieBPC")
```

- These instructions are also included in the Demo.R script on our GitHub repository: https://github.com/GENIE-BPC/intro_to_genieBPC_and_gnomeR
- Further R package details are available on the {genieBPC} GitHub repo & website
- {genieBPC} requires R version >=3.6

Clinico-Genomic Data Processing Pipeline

Data stored on SYNAPSE

Data import

`synapse_version()`

Indicates the versions of the data that are available to be specified in `pull_data_synapse()`

`pull_data_synapse()`

Imports GENIE BPC data from Synapse into the R environment

Data processing

`create_analytic_cohort()`

Selects an analytic cohort based on cancer diagnosis information and/or cancer-directed drug regimen information

`select_unique_ngs()`

Selects a unique next generation sequencing (NGS) test corresponding to the selected diagnoses

Data visualization

`drug_regimen_sunburst()`

Creates a sunburst figure of drug regimen information corresponding to the selected diagnoses in the order that the regimens were administered

Genomic Processing

{gnomeR}

create_gene_binary()

Processes data on mutation, CNA and fusions into analytic format

tbl_genomic()

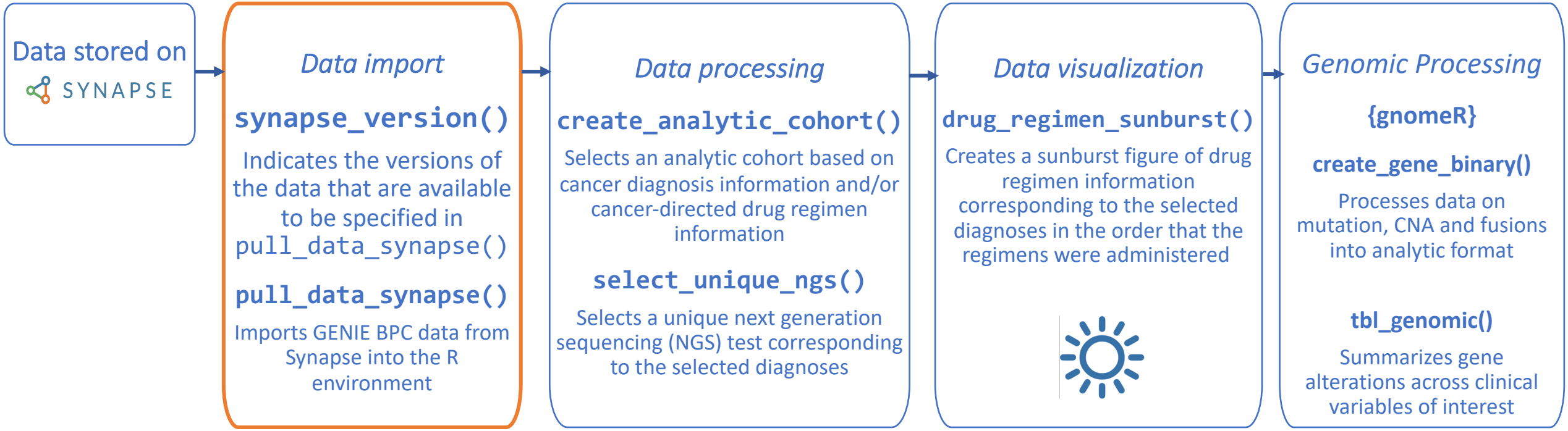
Summarizes gene alterations across clinical variables of interest

Case Study

Create a cohort of patients who were diagnosed with Stage IV adenocarcinoma non-small cell lung cancer (NSCLC) and received Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab as their first cancer-directed drug regimen after diagnosis.

Follow along using the Demo.R script on our GitHub repository: https://github.com/GENIE-BPC/intro_to_genieBPC_and_gnomeR

Clinico-Genomic Data Processing Pipeline



Set Synapse Credentials

To pull data from Synapse, users must create a Synapse account and store their Synapse credentials in the R environment. The **`set_synapse_credentials()`** function will store credentials during each R session:

```
set_synapse_credentials(username = 'your_username',
                        password = 'your_password')
```

Set Synapse Credentials

To pull data from Synapse, users must create a Synapse account and store their Synapse credentials in the R environment. The **`set_synapse_credentials()`** function will store credentials during each R session:

```
set_synapse_credentials(username = 'your_username',
                        password = 'your_password')
```

Coming soon

Additional functionality will be released soon to allow users to pass their Synapse Personal Access Token (PAT) through **`set_synapse_credentials()`**:

```
set_synapse_credentials(pat = 'your_pat')
```

`synapse_version()`

- Helper function that returns a table of GENIE BPC data releases that are currently available
- `synapse_version()` has one input: most_recent = TRUE/FALSE
  - Calling `genieBPC::synapse_version(most_recent = TRUE)` will return a table with each cancer cohort and its latest data release version
  - Calling `genieBPC::synapse_version(most_recent = FALSE)` will return a table with all cancer cohorts and data releases available

`synapse_version()`

- Helper function that returns a table of GENIE BPC data releases that are currently available
- `synapse_version()` has one input: most_recent = TRUE/FALSE
  - Calling `genieBPC::synapse_version(most_recent = TRUE)` will return a table with each cancer cohort and its latest data release version
  - Calling `genieBPC::synapse_version(most_recent = FALSE)` will return a table with all cancer cohorts and data releases available

`synapse_version(most_recent = TRUE)`

cohort	version	release_date	all_versions
BLADDER	v1.2-consortium	November 2023	Most Recent Versions
BrCa	v1.2-consortium	October 2022	Most Recent Versions
CRC	v1.2-consortium	August 2021	Most Recent Versions
CRC	v2.0-public	October 2022	Most Recent Versions
NSCLC	v2.1-consortium	August 2021	Most Recent Versions
NSCLC	v2.0-public	May 2022	Most Recent Versions
PANC	v1.2-consortium	January 2023	Most Recent Versions
Prostate	v1.2-consortium	January 2023	Most Recent Versions

`pull_data_synapse()`

- Pull GENIE BPC clinical and genomic data directly from Synapse into R
- Can specify cancer type (`cohort`) and version of data (`version`)
  - Version of the data is updated periodically on Synapse with re-releases (new variables available, additional QA, etc.)
- Returns a nested list of data frames for each cancer site for the accompanying version

Argument	Description	Acceptable Values
`cohort`	• GENIE BPC Project cancer • Currently, NSCLC and CRC are the only two publicly available datasets	• NSCLC • CRC • BrCa • PANC • Prostate • BLADDER
`version`	Version of the data (e.g v1.1-consortium, v2.0-public)	• Values can be found in `synapse_version()`

Demo: Run **`pull_data_synapse()`** for case study

Demo: Run **`pull_data_synapse()`** for case study

```
library(genieBPC)
```

Demo: Run **`pull_data_synapse()`** for case study

```
library(genieBPC)
set_synapse_credentials()
```

Demo: Run **`pull_data_synapse()`** for case study

```
library(genieBPC)
set_synapse_credentials()
nsclc_synapse_data <- pull_data_synapse(cohort = "NSCLC", version = "v2.0-public")
```

Demo: Run **`pull_data_synapse()`** for case study

```
library(genieBPC)

set_synapse_credentials()

nsclc_synapse_data <- pull_data_synapse(cohort = "NSCLC", version = "v2.0-public")
```

Calling **`nsclc_synapse_data$NSCLC_v2.0`** returns a list of datasets in `nsclc_synapse_data`:

- pt_char
- ca_dx_index
- ca_dx_non_index
- ca_drugs
- prissmm_pathology
- prissmm_imaging
- prissmm_md
- cpt
- mutations_extended
- cna
- fusions

Clinico-Genomic Data Processing Pipeline

Data stored on SYNAPSE

Data import

`synapse_version()`

Indicates the versions of the data that are available to be specified in `pull_data_synapse()`

`pull_data_synapse()`

Imports GENIE BPC data from Synapse into the R environment

Data processing

`create_analytic_cohort()`

Selects an analytic cohort based on cancer diagnosis information and/or cancer-directed drug regimen information

`select_unique_ngs()`

Selects a unique next generation sequencing (NGS) test corresponding to the selected diagnoses

Data visualization

`drug_regimen_sunburst()`

Creates a sunburst figure of drug regimen information corresponding to the selected diagnoses in the order that the regimens were administered

Genomic Processing

{gnomeR}

create_gene_binary()

Processes data on mutation, CNA and fusions into analytic format

tbl_genomic()

Summarizes gene alterations across clinical variables of interest

`create_analytic_cohort()`

Create a cohort from the GENIE BPC data

Cancer diagnosis information such as cancer cohort, treating institution, histology, and stage at diagnosis

Cancer-directed regimen information including regimen name and regimen order.

This function returns all clinical and genomic data for the selected patients

`create_analytic_cohort()`

Argument	Description	Acceptable Values
`data_synapse`	List returned from `pull_data_synapse()`	• Name of object in global environment that was returned from `pull_data_synapse()`

`create_analytic_cohort()`

Argument	Description	Acceptable Values
`data_synapse`	List returned from `pull_data_synapse()`	• Name of object in global environment that was returned from `pull_data_synapse()`
`index_ca_seq`	Index cancer sequence. Default is 1, indicating the patient's first index cancer. This is the cancer that met the eligibility criteria for the project and was selected at random for PRISSMM phenomic data curation.	• Numeric (1+)

`create_analytic_cohort()`

Argument	Description	Acceptable Values
`data_synapse`	List returned from `pull_data_synapse()`	• Name of object in global environment that was returned from `pull_data_synapse()`
`index_ca_seq`	Index cancer sequence. Default is 1, indicating the patient's first index cancer. This is the cancer that met the eligibility criteria for the project and was selected at random for PRISSMM phenomic data curation.	• Numeric (1+)
`institution`	GENIE BPC participating institution. Default selection is all institutions. *Note that not all institutions curated data for all cancer sites.*	• DFCI • MSK • UHN • VICC

create_analytic_cohort()

Argument	Description	Acceptable Values
stage_dx	Stage at diagnosis. Default selection is all stages.	• Stage I • Stage II • Stage III • Stage I-III NOS • Stage IV

create_analytic_cohort()

Argument	Description	Acceptable Values
stage_dx	Stage at diagnosis. Default selection is all stages.	• Stage I • Stage II • Stage III • Stage I-III NOS • Stage IV
histology	Cancer histology. Default selection is all histologies. For all cancer cohorts except for BrCa (breast cancer), this parameter corresponds to the variable 'ca_hist_adeno_squamous'. For BrCa, this parameter corresponds to the variable 'ca_hist_brca'	All cancer types except breast: • Adenocarcinoma • Squamous cell • Sarcoma • Small cell carcinoma • Other histologies/mixed tumor Breast cancer: • Invasive lobular carcinoma • Invasive ductal carcinoma • Other histology

create_analytic_cohort()

Argument	Description	Acceptable Values
regimen_drugs	Vector with names of drugs in cancer-directed regimen, separated by a comma. For example, to specify a regimen consisting of Carboplatin and Pemetrexed Disodium, specify regimen_drugs = "Carboplatin, Pemetrexed Disodium".	Acceptable values are found in the drug_names_by_cohort dataset provided with this package.
regimen_type	Indicates whether the regimen(s) specified in regimen_drugs indicates the exact regimen to return, or if regimens containing the drugs listed in regimen_drugs should be returned.	• Exact • Containing

Example: `regimen_drugs` and `regimen_type`

`regimen_drugs`	`regimen_type`	Example regimens returned
Carboplatin	Exact	• Carboplatin
Carboplatin	Containing	• Carboplatin • Carboplatin, Cisplatin • Carboplatin, Paclitaxel • Carboplatin, Pemetrexed Disodium • etc.

create_analytic_cohort()

Argument	Description	Acceptable Values
regimen_order	Order of cancer-directed regimen. If multiple drugs are specified, regimen_order indicates the regimen order for all drugs; different values of regimen_order cannot be specified for different drug regimens.	• Numeric (1+)
regimen_order_type	Specifies whether the 'regimen_order' parameter refers to the order of receipt of the drug regimen within the cancer diagnosis (across all other drug regimens; "within cancer") or the order of receipt of the drug regimen within the times that that drug regimen was administered (e.g. the first time carboplatin pemetrexed was received, out of all times that the patient received carboplatin pemetrexed; "within regimen").	• Within cancer • Within regimen

create_analytic_cohort()

Argument	Description	Acceptable Values
return_summary	Specifies whether summary tables are returned using {gtsummary}. Default is FALSE.	• TRUE • FALSE

Demo: `create_analytic_cohort()` for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

Demo: `create_analytic_cohort()` for case study using NSCLC 2.0-public data

```
nsclc_cohort <- create_analytic_cohort(
```

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
```

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
```

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma",
```

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma",
  regimen_drugs = c("Carboplatin, Pemetrexed Disodium",
```

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma",
  regimen_drugs = c("Carboplatin, Pemetrexed Disodium",
                    "Cisplatin, Pemetrexed Disodium",
```

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma",
  regimen_drugs = c("Carboplatin, Pemetrexed Disodium",
                    "Cisplatin, Pemetrexed Disodium",
                    "Bevacizumab, Carboplatin, Pemetrexed Disodium",
```

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma",
  regimen_drugs = c("Carboplatin, Pemetrexed Disodium",
                    "Cisplatin, Pemetrexed Disodium",
                    "Bevacizumab, Carboplatin, Pemetrexed Disodium",
                    "Bevacizumab, Cisplatin, Pemetrexed Disodium"),
```

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma",
  regimen_drugs = c("Carboplatin, Pemetrexed Disodium",
                    "Cisplatin, Pemetrexed Disodium",
                    "Bevacizumab, Carboplatin, Pemetrexed Disodium",
                    "Bevacizumab, Cisplatin, Pemetrexed Disodium"),
  regimen_type = "Exact",
```

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma",
  regimen_drugs = c("Carboplatin, Pemetrexed Disodium",
                    "Cisplatin, Pemetrexed Disodium",
                    "Bevacizumab, Carboplatin, Pemetrexed Disodium",
                    "Bevacizumab, Cisplatin, Pemetrexed Disodium"),
  regimen_type = "Exact",
  regimen_order = 1,
```

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma",
  regimen_drugs = c("Carboplatin, Pemetrexed Disodium",
                    "Cisplatin, Pemetrexed Disodium",
                    "Bevacizumab, Carboplatin, Pemetrexed Disodium",
                    "Bevacizumab, Cisplatin, Pemetrexed Disodium"),
  regimen_type = "Exact",
  regimen_order = 1,
  regimen_order_type = "within cancer",
```

Demo: **`create_analytic_cohort()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_cohort <- create_analytic_cohort(
  data_synapse = nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma",
  regimen_drugs = c("Carboplatin, Pemetrexed Disodium",
                    "Cisplatin, Pemetrexed Disodium",
                    "Bevacizumab, Carboplatin, Pemetrexed Disodium",
                    "Bevacizumab, Cisplatin, Pemetrexed Disodium"),
  regimen_type = "Exact",
  regimen_order = 1,
  regimen_order_type = "within cancer",
  return_summary = TRUE
)
```

nsclc_cohort $tbl_overall_ summary

Characteristic	N = 241 patients[1]
Number of diagnoses per patient in cohort_ca_dx data frame	
1	241 (100%)
Number of regimens per patient in cohort_ca_drugs data frame	
1	241 (100%)
Number of CPTs per patient in cohort_ngs data frame	
1	222 (92%)
2	18 (7.5%)
4	1 (0.4%)

[1] n (%)

nsclc_cohort
$tbl_cohort

Characteristic	N = 241 Diagnoses[1]
Cohort (cohort)	
NSCLC	241 (100%)
Institution (institution)	
DFCI	92 (38%)
MSK	118 (49%)
VICC	31 (13%)
Stage at diagnosis (stage_dx)	
Stage IV	241 (100%)
Histology (ca_hist_adeno_squamous)	
Adenocarcinoma	241 (100%)

[1] n (%)

nsclc_cohort $tbl_drugs

Characteristic	N = 241 Regimens[1]
Cohort (cohort)	
NSCLC	241 (100%)
Institution (institution)	
DFCI	92 (38%)
MSK	118 (49%)
VICC	31 (13%)
Drugs in regimen (regimen_drugs)	
Bevacizumab, Carboplatin, Pemetrexed Disodium	52 (22%)
Bevacizumab, Cisplatin, Pemetrexed Disodium	27 (11%)
Carboplatin, Pemetrexed Disodium	124 (51%)
Cisplatin, Pemetrexed Disodium	38 (16%)

[1] n (%)

nsclc_cohort $tbl_ngs

Characteristic	N = 262 Cancer Panel Tests[1]
Cohort (cohort)	
NSCLC	262 (100%)
Institution (institution)	
DFCI	99 (38%)
MSK	126 (48%)
VICC	37 (14%)
OncoTree code (cpt_oncotree_code)	
LCLC	1 (0.4%)
LUAD	253 (97%)
LUAS	1 (0.4%)
LUSC	1 (0.4%)
NSCLC	4 (1.5%)
NSCLCPD	2 (0.8%)
Sequence assay ID (cpt_seq_assay_id)	
DFCI-ONCOPANEL-1	1 (0.4%)
DFCI-ONCOPANEL-2	57 (22%)
DFCI-ONCOPANEL-3	41 (16%)
MSK-IMPACT341	3 (1.1%)
MSK-IMPACT410	61 (23%)
MSK-IMPACT468	62 (24%)
VICC-01-SOLIDTUMOR	26 (9.9%)
VICC-01-T5A	1 (0.4%)
VICC-01-T7	10 (3.8%)

[1] n (%)

Clinico-Genomic Data Processing Pipeline

Data stored on SYNAPSE

Data import

`synapse_version()`

Indicates the versions of the data that are available to be specified in `pull_data_synapse()`

`pull_data_synapse()`

Imports GENIE BPC data from Synapse into the R environment

Data processing

`create_analytic_cohort()`

Selects an analytic cohort based on cancer diagnosis information and/or cancer-directed drug regimen information

`select_unique_ngs()`

Selects a unique next generation sequencing (NGS) test corresponding to the selected diagnoses

Data visualization

`drug_regimen_sunburst()`

Creates a sunburst figure of drug regimen information corresponding to the selected diagnoses in the order that the regimens were administered

Genomic Processing

{gnomeR}

create_gene_binary()

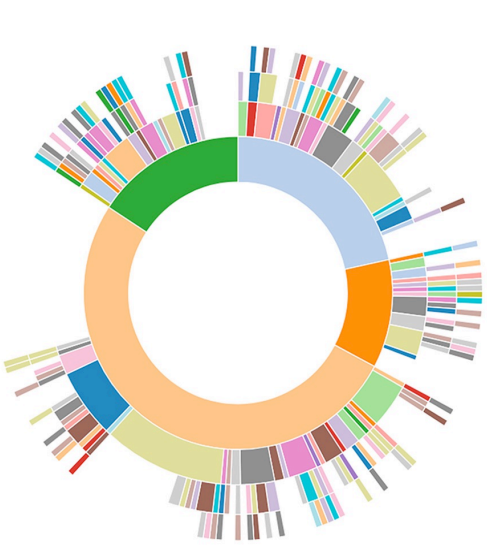
Processes data on mutation, CNA and fusions into analytic format

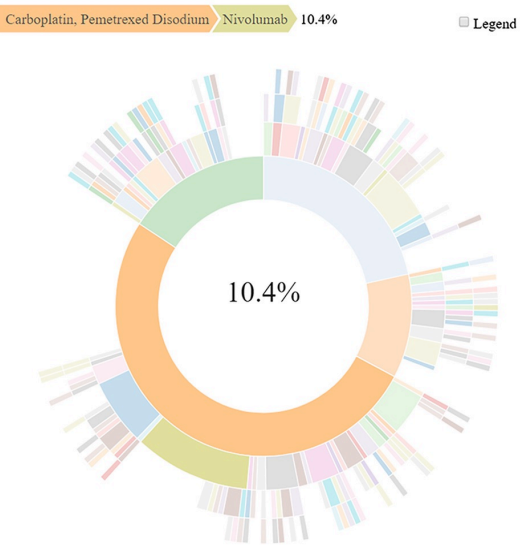
tbl_genomic()

Summarizes gene alterations across clinical variables of interest

drug_regimen_sunburst()

- Visualize the complete treatment course for selected cancer diagnoses
- Each ring corresponds to a regimen (i.e., innermost ring is first regimen, second innermost ring is second regimen, etc.)
- Interactive figure: Can hover to see regimen names and percent of patients receiving that regimen





drug_regimen_sunburst()

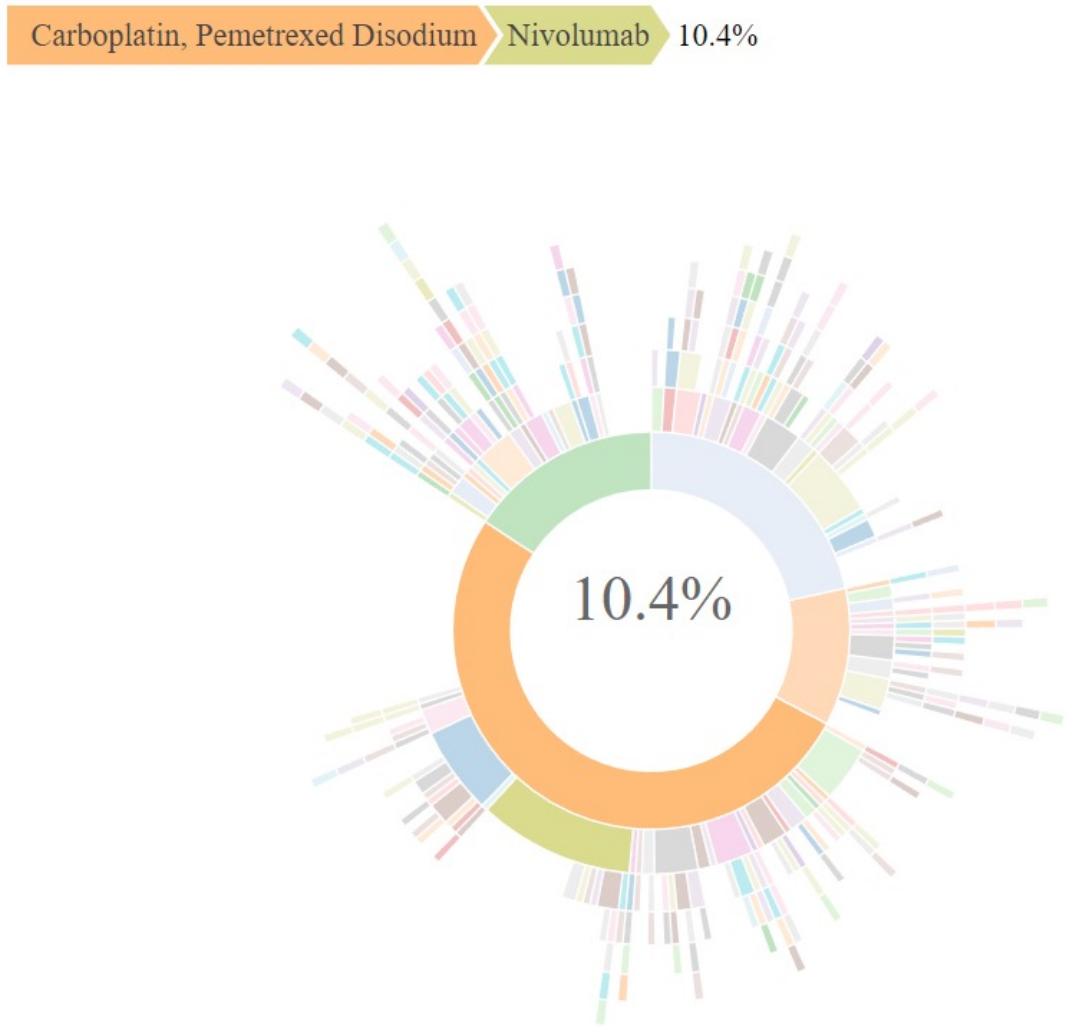
Argument	Description	Acceptable Values
data_synapse	List returned from pull_data_synapse()	• Name of object in global environment that was returned from pull_data_synapse()
data_cohort	The list returned from the create_analytic_cohort() function call	• Name of object in global environment that was returned from create_analytic_cohort()
max_n_regimens	The maximum number of regimens displayed in the sunburst plot	• Integer >0

Demo: **`drug_regimen_sunburst()`** for case study using NSCLC 2.0-public data

Case Study: Create a cohort of patients who were diagnosed with **Stage IV adenocarcinoma NSCLC** and received **Carboplatin and Pemetrexed +/- Bevacizumab or Cisplatin and Pemetrexed +/- Bevacizumab** as their **first** cancer-directed drug regimen after diagnosis

```
nsclc_sunburst <- drug_regimen_sunburst(
            data_synapse = nsclc_synapse_data$NSCLC_v2.0,
            data_cohort = nsclc_cohort)
```

nsclc_sunburst$
sunburst_plot



Clinico-Genomic Data Processing Pipeline

Data stored on SYNAPSE

Data import

`synapse_version()`

Indicates the versions of the data that are available to be specified in `pull_data_synapse()`

`pull_data_synapse()`

Imports GENIE BPC data from Synapse into the R environment

Data processing

`create_analytic_cohort()`

Selects an analytic cohort based on cancer diagnosis information and/or cancer-directed drug regimen information

`select_unique_ngs()`

Selects a unique next generation sequencing (NGS) test corresponding to the selected diagnoses

Data visualization

`drug_regimen_sunburst()`

Creates a sunburst figure of drug regimen information corresponding to the selected diagnoses in the order that the regimens were administered

Genomic Processing

{gnomeR}

create_gene_binary()

Processes data on mutation, CNA and fusions into analytic format

tbl_genomic()

Summarizes gene alterations across clinical variables of interest

Case Study

Using the cohort of patients who were diagnosed with Stage IV adenocarcinoma NSCLC, we will:

1. **Process data** into a analysis-ready matrix of gene alteration events
2. **Summarize genomic alteration frequencies** and analyze differences between males and females

```r
nsclc_cohort <- create_analytic_cohort(
  data_synapse =
    nsclc_synapse_data$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma",
  regimen_drugs =
    c("Carboplatin, Pemetrexed Disodium",
      "Cisplatin, Pemetrexed Disodium",
      "Bevacizumab, Carboplatin, Pemetrexed Disodium",
      "Bevacizumab, Cisplatin, Pemetrexed Disodium"),
  regimen_type = "Exact",
  regimen_order = 1,
  regimen_order_type = "within cancer",
  return_summary = TRUE
)
```

Characteristic	N = 241[1]
naaccr_sex_code	
Female	145 (60%)
Male	96 (40%)

[1] n (%)

Overview of Genomic Data

We will be processing and analyzing data on:

1) Mutations

```
mutations <- nsclc_synapse_data$NSCLC_v2.0$mutations_extended
```

2) Discrete Copy Number Alterations

```
cna <- nsclc_synapse_data$NSCLC_v2.0$cna
```

3) Fusions

```
fusions <- nsclc_synapse_data$NSCLC_v2.0$fusions
```

AACR Data Guide: https://www.aacr.org/wp-content/uploads/2020/02/20200127_GENIE_Data_Guide_7.pdf

Processing Data

Issues When Processing Multi-Institutional Genomic Data

Issues When Processing Multi-Institutional Genomic Data

1. **Multiple Samples Per Patient**

Issues When Processing Multi-Institutional Genomic Data

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

Issues When Processing Multi-Institutional Genomic Data

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

2. **Data Formats & Gene Standards Often Inconsistent**

Issues When Processing Multi-Institutional Genomic Data

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

2. **Data Formats & Gene Standards Often Inconsistent**
    - Column names, data formats and gene names may differ between studies or even within studies

Issues When Processing Multi-Institutional Genomic Data

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

2. **Data Formats & Gene Standards Often Inconsistent**
    - Column names, data formats and gene names may differ between studies or even within studies

3. **Cohort Inclusion**

Issues When Processing Multi-Institutional Genomic Data

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

2. **Data Formats & Gene Standards Often Inconsistent**
    - Column names, data formats and gene names may differ between studies or even within studies

3. **Cohort Inclusion**
    - Samples with no alterations may be dropped when pulling data

Issues When Processing Multi-Institutional Genomic Data

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

2. **Data Formats & Gene Standards Often Inconsistent**
    - Column names, data formats and gene names may differ between studies or even within studies

3. **Cohort Inclusion**
    - Samples with no alterations may be dropped when pulling data

4. **Multi-Institutional Studies Use Several Gene Panels**

Issues When Processing Multi-Institutional Genomic Data

1. **Multiple Samples Per Patient**
   - We often need to select a single sample per patient for analyses

2. **Data Formats & Gene Standards Often Inconsistent**
   - Column names, data formats and gene names may differ between studies or even within studies

3. **Cohort Inclusion**
   - Samples with no alterations may be dropped when pulling data

4. **Multi-Institutional Studies Use Several Gene Panels**
   - Samples may be sequenced using different panels, therefore the non-overlapping genes have to be annotated as missing

Select One Sample Per Patient

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

Argument	Description
data_cohort	Output object of the create_analytic_cohort function.
oncotree_code	Character vector specifying which sample OncoTree codes to keep. See 'cpt_oncotree_code' column of data_cohort.
sample_type	Character specifying which type of genomic sample to prioritize. Options are 'Primary', 'Local', and 'Metastasis'. Default is to not select a NGS sample based on the sample type.
min_max_time	Character specifying if the first or last genomic sample recorded should be kept. Options are 'min' (first) and 'max' (last).

Select One Sample Per Patient

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

- Some patients have multiple sequencing reports (e.g. pre & post treatment samples, primary and metastatic samples)

Argument	Description
data_cohort	Output object of the create_analytic_cohort function.
oncotree_code	Character vector specifying which sample OncoTree codes to keep. See 'cpt_oncotree_code' column of data_cohort.
sample_type	Character specifying which type of genomic sample to prioritize. Options are 'Primary', 'Local', and 'Metastasis'. Default is to not select a NGS sample based on the sample type.
min_max_time	Character specifying if the first or last genomic sample recorded should be kept. Options are 'min' (first) and 'max' (last).

Select One Sample Per Patient

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

- Some patients have multiple sequencing reports (e.g. pre & post treatment samples, primary and metastatic samples)
- Two approaches to analysis:

Argument	Description
data_cohort	Output object of the create_analytic_cohort function.
oncotree_code	Character vector specifying which sample OncoTree codes to keep. See 'cpt_oncotree_code' column of data_cohort.
sample_type	Character specifying which type of genomic sample to prioritize. Options are 'Primary', 'Local', and 'Metastasis'. Default is to not select a NGS sample based on the sample type.
min_max_time	Character specifying if the first or last genomic sample recorded should be kept. Options are 'min' (first) and 'max' (last).

Select One Sample Per Patient

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

- Some patients have multiple sequencing reports (e.g. pre & post treatment samples, primary and metastatic samples)
- Two approaches to analysis:
    1. Aggregate information across sequencing reports (e.g. “ever” altered)

Argument	Description
data_cohort	Output object of the create_analytic_cohort function.
oncotree_code	Character vector specifying which sample OncoTree codes to keep. See ‘cpt_oncotree_code’ column of data_cohort.
sample_type	Character specifying which type of genomic sample to prioritize. Options are ‘Primary’, ‘Local’, and ‘Metastasis’. Default is to not select a NGS sample based on the sample type.
min_max_time	Character specifying if the first or last genomic sample recorded should be kept. Options are ‘min’ (first) and ‘max’ (last).

Select One Sample Per Patient

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

- Some patients have multiple sequencing reports (e.g. pre & post treatment samples, primary and metastatic samples)
- Two approaches to analysis:
    1. Aggregate information across sequencing reports (e.g. “ever” altered)
    2. Select one unique sequencing report -> **We focus on this approach**

Argument	Description
data_cohort	Output object of the create_analytic_cohort function.
oncotree_code	Character vector specifying which sample OncoTree codes to keep. See ‘cpt_oncotree_code’ column of data_cohort.
sample_type	Character specifying which type of genomic sample to prioritize. Options are ‘Primary’, ‘Local’, and ‘Metastasis’. Default is to not select a NGS sample based on the sample type.
min_max_time	Character specifying if the first or last genomic sample recorded should be kept. Options are ‘min’ (first) and ‘max’ (last).

Select One Sample Per Patient

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

- Some patients have multiple sequencing reports (e.g. pre & post treatment samples, primary and metastatic samples)
- Two approaches to analysis:
    1. Aggregate information across sequencing reports (e.g. “ever” altered)
    2. Select one unique sequencing report -> **We focus on this approach**
- We can use `select_unique_ngs()` to select 1 sample per patient

Argument	Description
data_cohort	Output object of the create_analytic_cohort function.
oncotree_code	Character vector specifying which sample OncoTree codes to keep. See ‘cpt_oncotree_code’ column of data_cohort.
sample_type	Character specifying which type of genomic sample to prioritize. Options are ‘Primary’, ‘Local’, and ‘Metastasis’. Default is to not select a NGS sample based on the sample type.
min_max_time	Character specifying if the first or last genomic sample recorded should be kept. Options are ‘min’ (first) and ‘max’ (last).

Select One Sample Per Patient

1. **Multiple Samples Per Patient**
    - We often need to select a single sample per patient for analyses

```
nrow(nsclc_cohort$cohort_ngs)
```

[1] 262

```
nsclc_samp <- select_unique_ngs(
  data_cohort = nsclc_cohort$cohort_ngs,
  oncotree_code = "LUAD",
  sample_type = "Metastasis",
  min_max_time = "max"
)

nrow(nsclc_samp)
```

[1] 241

Format Data in Analysis-ready Matrix

sample_id	CREBBP	GLI2	KRAS	MAP3K1	PIK3C2B	PBRM1	MYC.Del
GENIE-DFCI-004022-1313	0	0	0	0	0	NA	0
GENIE-DFCI-000013-8840	0	0	1	0	0	0	0
GENIE-DFCI-000136-6004	0	0	1	0	0	0	0
GENIE-DFCI-000215-8010	0	0	0	0	0	0	0
GENIE-DFCI-000381-9526	0	0	0	0	0	0	0

Format Data in Analysis-ready Matrix

- Next we want to get our genomic data in an **analysis-friendly format**.

sample_id	CREBBP	GLI2	KRAS	MAP3K1	PIK3C2B	PBRM1	MYC.Del
GENIE-DFCI-004022-1313	0	0	0	0	0	NA	0
GENIE-DFCI-000013-8840	0	0	1	0	0	0	0
GENIE-DFCI-000136-6004	0	0	1	0	0	0	0
GENIE-DFCI-000215-8010	0	0	0	0	0	0	0
GENIE-DFCI-000381-9526	0	0	0	0	0	0	0

Format Data in Analysis-ready Matrix

- Next we want to get our genomic data in an **analysis-friendly format**.
- `create_gene_binary()` from {gnomeR} will give us a data frame of ***n* patients x *p* alterations**.

sample_id	CREBBP	GLI2	KRAS	MAP3K1	PIK3C2B	PBRM1	MYC.Del
GENIE-DFCI-004022-1313	0	0	0	0	0	NA	0
GENIE-DFCI-000013-8840	0	0	1	0	0	0	0
GENIE-DFCI-000136-6004	0	0	1	0	0	0	0
GENIE-DFCI-000215-8010	0	0	0	0	0	0	0
GENIE-DFCI-000381-9526	0	0	0	0	0	0	0

Format Data in Analysis-ready Matrix

- Next we want to get our genomic data in an **analysis-friendly format**.
- `create_gene_binary()` from {gnomeR} will give us a data frame of ***n* patients x *p* alterations**.
- Alteration columns are denoted by the **gene name** if mutation (e.g. `TP53`) or **gene name + .Amp, .fus, .Del** (`TP53.Del`) for other alterations types.

sample_id	CREBBP	GLI2	KRAS	MAP3K1	PIK3C2B	PBRM1	MYC.Del
GENIE-DFCI-004022-1313	0	0	0	0	0	NA	0
GENIE-DFCI-000013-8840	0	0	1	0	0	0	0
GENIE-DFCI-000136-6004	0	0	1	0	0	0	0
GENIE-DFCI-000215-8010	0	0	0	0	0	0	0
GENIE-DFCI-000381-9526	0	0	0	0	0	0	0

Format Data in Analysis-ready Matrix

- Next we want to get our genomic data in an **analysis-friendly format**.
- `create_gene_binary()` from {gnomeR} will give us a data frame of ***n* patients x *p* alterations**.
- Alteration columns are denoted by the **gene name** if mutation (e.g. `TP53`) or **gene name + .Amp, .fus, .Del** (`TP53.Del`) for other alterations types.
- Each cell will have `0` if no alteration, `1` if altered, or `NA` if that gene was not tested in that patient.

sample_id	CREBBP	GLI2	KRAS	MAP3K1	PIK3C2B	PBRM1	MYC.Del
GENIE-DFCI-004022-1313	0	0	0	0	0	NA	0
GENIE-DFCI-000013-8840	0	0	1	0	0	0	0
GENIE-DFCI-000136-6004	0	0	1	0	0	0	0
GENIE-DFCI-000215-8010	0	0	0	0	0	0	0
GENIE-DFCI-000381-9526	0	0	0	0	0	0	0

Get Data in Standardized Format

2. **Data Formats & Gene Standards Often Inconsistent**
    - Column names, data formats and gene names may differ between studies or even within studies!

{gnomeR} functions are designed to work with standard alteration data formats used in common platforms like cBioPortal.

GENIE CNA and fusion data are slightly different than this standard, therefore we need to reformat it using:

- `gnomeR::reformat_fusions()`
- `gnomeR::pivot_cna_longer()`

Documentation on data formats: https://docs.cbioportal.org/file-formats/

Get Data in Standardized Format

Format fusions to follow the cBioPortal standard:

Before:

```
1 fusions
```

Hugo_Symbol	Tumor_Sample_Barcode	Fusion
TP53	GENIE-MSK-P-0004827-T01-IM5	TP53-intragenic
XPO1	GENIE-MSK-P-0004827-T01-IM5	XPO1-USP34 fusion
USP34	GENIE-MSK-P-0004827-T01-IM5	XPO1-USP34 fusion

After:

```
1 reformat_fusion(fusions)
```

sample_id	site_1_hugo_symbol	site_2_hugo_symbol	fusion
GENIE-MSK-P-0004827-T01-IM5	TP53	NA	TP53
GENIE-MSK-P-0004827-T01-IM5	USP34	XPO1	USP34-XPO1

Get Data in Standardized Format

Format CNA to follow the cBioPortal standard:

Before:

```
1 cna
```

Hugo_Symbol	GENIE.DFCI.003908.234520	GENIE.DFCI.002183.6917
APC	1	0
ARID1B	-1	0
BCL2	-1	0
TP53	0	0

After:

```
1 gnomeR::pivot_cna_longer(cna)
```

hugo_symbol	sample_id	alteration
APC	GENIE-DFCI-003908-234520	gain
ARID1B	GENIE-DFCI-003908-234520	loss
BCL2	GENIE-DFCI-003908-234520	loss
BCL6	GENIE-DFCI-002183-6917	gain
CARD11	GENIE-DFCI-002183-6917	gain

Documentation on data formats: https://docs.cbioportal.org/file-formats/

Format Data in Analysis-ready Matrix

Basic code:

```
gnomeR::create_gene_binary(
  mutation = mutations,
  cna = reformat_cna,
  fusion = reformat_fusions)
```

Format Data in Analysis-ready Matrix

- Data is now in standardized format and can be processed using {gnomeR} `create_gene_binary()`

Basic code:

```
gnomeR::create_gene_binary(
  mutation = mutations,
  cna = reformat_cna,
  fusion = reformat_fusions)
```

Format Data in Analysis-ready Matrix

- Data is now in standardized format and can be processed using {gnomeR} `create_gene_binary()`
- We will add additional arguments to `create_gene_binary()` to help address remaining data processing issues

Basic code:

```
gnomeR::create_gene_binary(
  mutation = mutations,
  cna = reformat_cna,
  fusion = reformat_fusions)
```

Cohort Inclusion

3. **Cohort Inclusion**
    - Samples with no alterations may be dropped when pulling raw genomic data

```
gnomeR::create_gene_binary(
  mutation = mutations,
  cna = reformat_cna,
  fusion = reformat_fusions
  samples = nsclc_samp$cpt_genie_sample_id)
```

The `samples` argument will ensure all study IDs have a row in resulting analysis data, even if they are not present in genomic files

Not All Patients Tested on Same Panel

4. **Multi-Institutional Studies Use Several Gene Panels**
    - Samples may be sequenced using different panels therefore the non overlapping genes have to be annotated as missing

```
gnomeR::create_gene_binary(
  mutation = mutations,
  cna = reformat_cna,
  fusion = reformat_fusions,
  samples = nsclc_samp$cpt_genie_sample_id,
  specify_panel = nsclc_panels)
```

The `specify_panels` argument can insert `NA`s when we know that gene was not tested for a specific set of patients.

Not All Patients Tested on Same Panel

To use `specify_panels`, we first need to create a data frame indicating which patient IDs were sequenced on which panels.

```
nsclc_panels <- data.frame(
  sample_id = nsclc_samp$cpt_genie_sample_id,
  panel_id = nsclc_samp$cpt_seq_assay_id) %>%
  mutate(panel_id = ifelse(!is.na(panel_id),
                           panel_id, "no"))

nsclc_panels %>% head()
```

sample_id	panel_id
GENIE-DFCI-000013-8840	DFCI-ONCOPANEL-2
GENIE-DFCI-000136-6004	DFCI-ONCOPANEL-2
GENIE-DFCI-000215-8010	DFCI-ONCOPANEL-2
GENIE-DFCI-000381-9526	DFCI-ONCOPANEL-2
GENIE-DFCI-000410-10003	DFCI-ONCOPANEL-2
GENIE-DFCI-000583-11175	DFCI-ONCOPANEL-2

Not All Patients Tested on Same Panel

Without Panel Annotation

```
binmat1 <- gnomeR::create_gene_binary(
  mutation = mutations,
  cna = reformat_cna,
  fusion = reformat_fusions,
  samples = nsclc_samp$cpt_genie_sample_id,
  specify_panel = "no")
```

sample_id	panel_id	GLI2	KRAS	PIK3C2B	PBRM1
GENIE-DFCI-004022-1313	DFCI-ONCOPANEL-1	0	0	0	0
GENIE-DFCI-000013-8840	DFCI-ONCOPANEL-2	0	1	0	0
GENIE-MSK-P-0002725-T01-IM3	MSK-IMPACT341	0	1	0	0
GENIE-MSK-P-0017722-T02-IM6	MSK-IMPACT468	0	0	0	0

With Panel Annotation

```
binmat2 <- gnomeR::create_gene_binary(
  mutation = mutations,
  cna = reformat_cna,
  fusion = reformat_fusions,
  samples = nsclc_samp$cpt_genie_sample_id,
  specify_panel = nsclc_panels)
```

sample_id	panel_id	GLI2	KRAS	PIK3C2B	PBRM1
GENIE-DFCI-004022-1313	DFCI-ONCOPANEL-1	0	0	0	NA
GENIE-DFCI-000013-8840	DFCI-ONCOPANEL-2	0	1	0	0
GENIE-MSK-P-0002725-T01-IM3	MSK-IMPACT341	NA	1	NA	0
GENIE-MSK-P-0017722-T02-IM6	MSK-IMPACT468	NA	0	NA	0

Ensure Gene Names Are Consistent Across Studies

2. **Data Formats & Gene Standards Often Inconsistent**
    - Column names, data formats and gene names may differ between studies or even within studies.

```
no_recode <- gnomeR::create_gene_binary(
  samples = nsclc_samp$cpt_genie_sample_id,
  mutation = mutations,
  cna = reformat_cna,
  fusion = reformat_fusions,
  specify_panel = nsclc_panels,
  recode_aliases = "no")
```

```
recode <- gnomeR::create_gene_binary(
  samples = nsclc_samp$cpt_genie_sample_id,
  mutation = mutations,
  cna = reformat_cna,
  fusion = reformat_fusions,
  specify_panel = nsclc_panels,
  recode_aliases = "impact")
```

```
setdiff(names(no_recode), names(recode)) %>% head()
```

[1] “MRE11A” “RFWD2” “H3F3A” “FAM46C” “HIST1H3D” “WHSC1L1”

Process Data: Final Dataset

Let's run `create_gene_binary()` with the `samples`, `specify_panel` and `recode_aliases` arguments.

First we create `nsclc_panels`:

```
nsclc_panels <- data.frame(
  sample_id = nsclc_samp$cpt_genie_sample_id,
  panel_id = nsclc_samp$cpt_seq_assay_id) %>%
  mutate(panel_id = ifelse(!is.na(panel_id),
                           panel_id, "no"))
```

Then run `create_gene_binary()`:

```
gene_binary <- gnomeR::create_gene_binary(
  mutation = mutations,
  cna = reformat_cna,
  fusion = reformat_fusions,
  samples = nsclc_samp$cpt_genie_sample_id,
  specify_panel = nsclc_panels,
  recode_aliases = "impact")
```

Analyzing Data

Issues When Analyzing Multi-Institutional Genomic Data

Issues When Analyzing Multi-Institutional Genomic Data

1. **Multiple Testing and False Positives**

Issues When Analyzing Multi-Institutional Genomic Data

1. **Multiple Testing and False Positives**

   - Many hypothesis tests done simultaneously can lead to false positive findings.

Issues When Analyzing Multi-Institutional Genomic Data

1. **Multiple Testing and False Positives**

- Many hypothesis tests done simultaneously can lead to false positive findings.
- Very low prevalence genes are often not very informative.

Issues When Analyzing Multi-Institutional Genomic Data

1. **Multiple Testing and False Positives**

- Many hypothesis tests done simultaneously can lead to false positive findings.
- Very low prevalence genes are often not very informative.
- Choose a threshold (e.g. 1% or 5 %) *a priori* and consider reporting a q-value (adjusted for multiple testing).

Issues When Analyzing Multi-Institutional Genomic Data

1. **Multiple Testing and False Positives**

- Many hypothesis tests done simultaneously can lead to false positive findings.
- Very low prevalence genes are often not very informative.
- Choose a threshold (e.g. 1% or 5 %) *a priori* and consider reporting a q-value (adjusted for multiple testing).

2. **Limited Power To Detect Clinical Associations When Sparse Alterations**

Issues When Analyzing Multi-Institutional Genomic Data

1. **Multiple Testing and False Positives**

- Many hypothesis tests done simultaneously can lead to false positive findings.
- Very low prevalence genes are often not very informative.
- Choose a threshold (e.g. 1% or 5 %) *a priori* and consider reporting a q-value (adjusted for multiple testing).

2. **Limited Power To Detect Clinical Associations When Sparse Alterations**

- If biologically meaningful, you may want to summarize on gene or pathway level

Case Study

Case Study

- We will use the processed binary data frame data (`gene_binary`) to **summarize genomic alterations** overall in the cohort, and **by sex**.

Case Study

- We will use the processed binary data frame data (`gene_binary`) to **summarize genomic alterations** overall in the cohort, and **by sex**.

- First, we need to join clinical data on `sex` to genomic data

Add Clinical Variable To Data

```r
# get patient IDs and sample IDs
patient_index <-  nsclc_cohort$cohort_ngs %>%
  select(record_id, cpt_genie_sample_id)

# Join sex data to patient ID index
select_clinical <- nsclc_cohort$cohort_pt_char %>%
  select(record_id, naaccr_sex_code) %>%
  left_join(patient_index)

# Join all to gene binary data
gene_binary <- gene_binary %>%
  left_join(select_clinical,
            by = c("sample_id"= "cpt_genie_sample_id")) %>%
  select(-record_id)

gene_binary <- gene_binary %>%
  select(sample_id, naaccr_sex_code, everything())

gene_binary %>%
  select(naaccr_sex_code) %>%
  tbl_summary()
```

Characteristic	N = 241[1]
naaccr_sex_code	
Female	145 (60%)
Male	96 (40%)

[1] n (%)

Subset By a Prevalence Threshold

1. **Multiple Testing and False Positives**

- Use `subset_by_frequency(t)` to **subset genes** above a given threshold
- `t` indicates a **prevalence threshold** between 0 (`t = 0`) and 100% (`t = 1`)
- `other_vars` retains the clinical variable of interest in the resulting data set

Subset By 40% Threshold

```
ncol(gene_binary)
```

[1] 1403

```
nsclc_subset <- gene_binary %>%
  subset_by_frequency(t = .4, other_vars = naaccr_sex_code)

ncol(nsclc_subset)
```

[1] 6

Subset by a Panel

Use `subset_by_panel()` to subset genes in a given targeted panel.

```
ncol(gene_binary)
```

[1] 1403

```
nsclc_subset_panel <- gene_binary %>%
  subset_by_panel(panel_id = 'IMPACT300', other_vars = naaccr_sex_code)

ncol(nsclc_subset_panel)
```

[1] 220

Summarize Alterations with tbl_genomic()

- `tbl_genomic()` is a wrapper function for `gtsummary::tbl_summary()` specifically designed for presenting genomic data
- You can use any **{gtsummary}** function on top of `tbl_genomic()` to customize the table (e.g. `bold_labels()`)

Summarize Alterations with tbl_genomic

Create a simple `tbl_genomic` object, then bold the labels.

```
nsclc_subset %>%
  select(-naaccr_sex_code) %>%
  tbl_genomic() %>%
  bold_labels()
```

Characteristic	N = 241[1]
TP53	124 (51%)
GBA.Amp	17 (46%)
Unknown	204
KRAS	98 (41%)
JAZF1.Amp	15 (41%)
Unknown	204

[1] n (%)

Summarize Genes with tbl_genomic()

You may want to analyze on the **gene level** instead of the alteration level.

Use `summarize_by_gene()` first, then pass to `tbl_genomic()`:

> ⚠ **Note: `summarize_by_gene()` should come before passing to `subset_by_frequency()`**

```
1 tbl_gene <- gene_binary %>%
2   select(-naaccr_sex_code) %>%
3   summarize_by_gene() %>%
4   subset_by_frequency(t = .4) %>%
5   tbl_genomic()
6
7 tbl_gene
```

Characteristic	N = 241[1]
TP53	135 (56%)
GBA	17 (46%)
Unknown	204
PTK2B	17 (46%)
Unknown	204
KRAS	107 (44%)
WRN	38 (41%)
Unknown	149
JAZF1	15 (41%)
Unknown	204

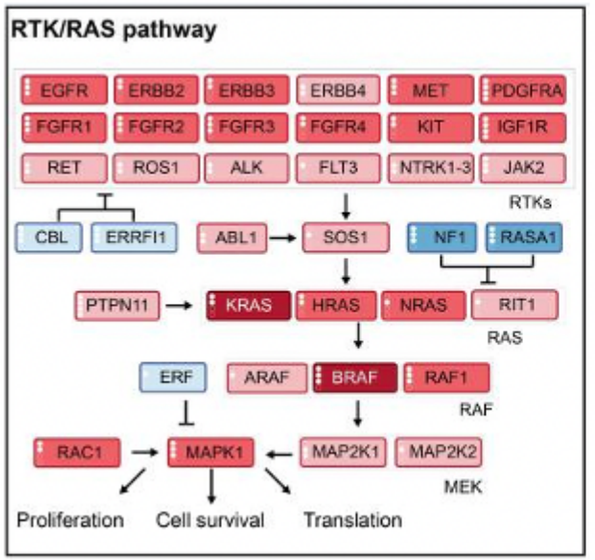
[1] n (%)

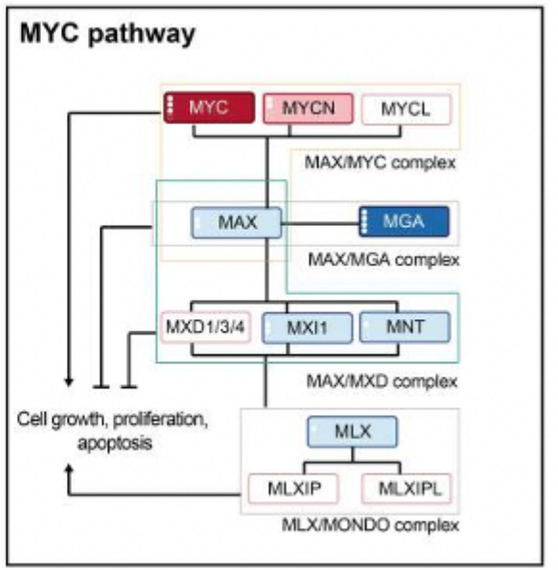
Summarize Oncogenic Pathways

2. **Limited Power To Detect Clinical Associations When Sparse Alterations**

- If biologically meaningful, you may want to summarize on Alteration, Gene or Pathway Level

Often we want to analyze alterations on the pathway level.





Sanchez-Vega F, et al. Oncogenic Signaling Pathways in The Cancer Genome Atlas. Cell. 2018 Apr 5;173(2):321-337.e10. doi:

Summarize Oncogenic Pathways

2. **Limited Power To Detect Clinical Associations When Sparse Alterations**

- If biologically meaningful, you may want to summarize on Alteration, Gene or Pathway Level

{gnomeR} offers several default pathways that can be added with `add_pathways()`

```
1 paths <- gnomeR::pathways %>% names()
2 paths
```

[1] “RTK/RAS” “Nrf2” “PI3K” “TGFB” “p53”
[6] “Wnt” “Myc” “Cell cycle” “Hippo” “Notch”

Summarize Oncogenic Pathways

You can also add a custom pathway:

```
path_df <- gene_binary %>%
  add_pathways(custom_pathways = c("SPOP.mut", "FOXA1.mut"))

path_df %>% select("pathway_custom") %>%
  tbl_summary()
```

Characteristic	N = 241[1]
pathway_custom	2 (0.8%)

[1] n (%)

Note: You must specify `.mut`, `.Amp`, `.Del` for alterations `custom_pathways`

Summarize Alteration Pathways

```
path_df <- gene_binary %>%
  select(-naaccr_sex_code) %>%
  add_pathways()

path_df %>%
  select(contains("pathway")) %>%
  tbl_summary() %>%
  bold_labels()
```

Summarize Alteration Pathways

```r
path_df <- gene_binary %>%
  select(-naaccr_sex_code) %>%
  add_pathways()

path_df %>%
  select(contains("pathway")) %>%
  tbl_summary() %>%
  bold_labels()
```

Characteristic	N = 241[1]
pathway_RTK/RAS	217 (90%)
pathway_Nrf2	48 (20%)
pathway_PI3K	114 (47%)
pathway_TGFB	34 (14%)
pathway_p53	181 (75%)
pathway_Wnt	51 (21%)
pathway_Myc	70 (29%)
pathway_Cell cycle	90 (37%)
pathway_Hippo	31 (13%)
pathway_Notch	72 (30%)

[1] n (%)

Comparing Alteration Frequencies Across Clinical Data

We can easily compare frequencies by sex using the by argument:

```
tbl_gene <- gene_binary %>%
  subset_by_frequency(
    t = .4,
    other_vars = naaccr_sex_code) %>%
  tbl_genomic(by = naaccr_sex_code) %>%
  bold_labels()

tbl_gene
```

Comparing Alteration Frequencies Across Clinical Data

We can easily compare frequencies by sex using the by argument:

```
tbl_gene <- gene_binary %>%
  subset_by_frequency(
    t = .4,
    other_vars = naaccr_sex_code) %>%
  tbl_genomic(by = naaccr_sex_code) %>%
  bold_labels()

tbl_gene
```

Characteristic	**Overall**, N = 241[1]	**Female**, N = 145[1]	**Male**, N = 96[1]
TP53	124 (51%)	72 (50%)	52 (54%)
GBA.Amp	17 (46%)	8 (36%)	9 (60%)
Unknown	204	123	81
KRAS	98 (41%)	66 (46%)	32 (33%)
JAZF1.Amp	15 (41%)	9 (41%)	6 (40%)
Unknown	204	123	81

[1] n (%)

Comparing Alteration Frequencies Across Clinical Data

1. **Multiple Testing and False Positives**

We can use {gtsummary}'s `add_p()` and `add_q()` for hypothesis testing

```
tbl_gene <- gene_binary %>%
  subset_by_frequency(
    t = .4,
    other_vars = naaccr_sex_code) %>%
  tbl_genomic(
    by = naaccr_sex_code) %>%
  bold_labels() %>%
  add_p() %>%
  add_q()

tbl_gene
```

Characteristic	Overall, N = 241[1]	Female, N = 145[1]	Male, N = 96[1]	p-value[2]	q-value[3]
TP53	124 (51%)	72 (50%)	52 (54%)	0.5	0.7
GBA.Amp	17 (46%)	8 (36%)	9 (60%)	0.2	0.3
Unknown	204	123	81		
KRAS	98 (41%)	66 (46%)	32 (33%)	0.059	0.2
JAZF1.Amp	15 (41%)	9 (41%)	6 (40%)	>0.9	>0.9
Unknown	204	123	81		

[1] n (%)

[2] Pearson's Chi-squared test

[3] False discovery rate correction for multiple testing

Comparing Alteration Frequencies Across Clinical Data

```r
tbl_gene <- gene_binary %>%
  subset_by_frequency(
    t = .4,
    other_vars = naaccr_sex_code) %>%
  tbl_genomic(by = naaccr_sex_code) %>%
  bold_labels() %>%
  add_p() %>%
  add_q()

tbl_gene
```

Characteristic	**Overall**, N = 241[1]	**Female**, N = 145[1]	**Male**, N = 96[1]	**p-value**[2]	**q-value**[3]
TP53	124 (51%)	72 (50%)	52 (54%)	0.5	0.7
GBA.Amp	17 (46%)	8 (36%)	9 (60%)	0.2	0.3
Unknown	204	123	81		
KRAS	98 (41%)	66 (46%)	32 (33%)	0.059	0.2
JAZF1.Amp	15 (41%)	9 (41%)	6 (40%)	>0.9	>0.9
Unknown	204	123	81		

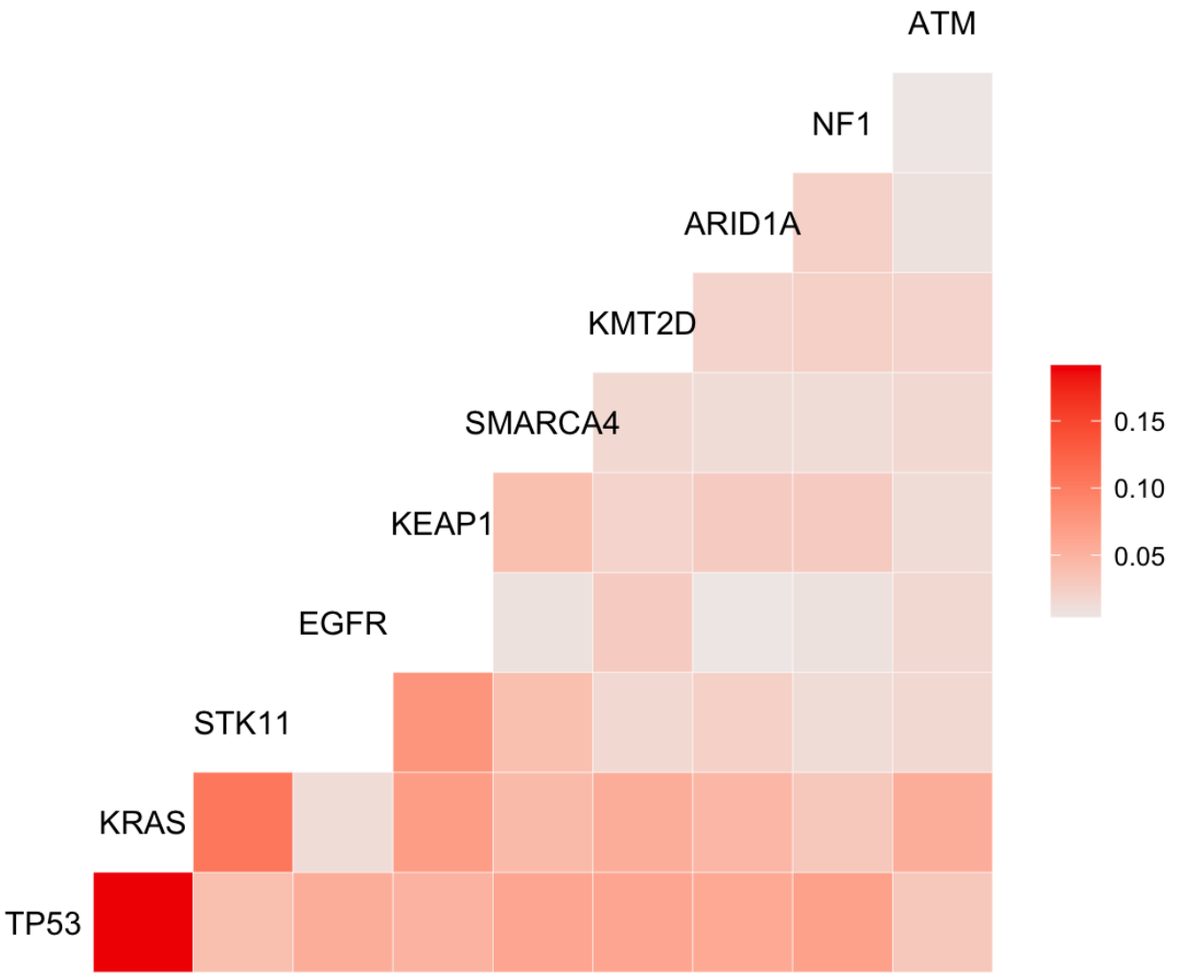
[1] n (%)

[2] Pearson's Chi-squared test

[3] False discovery rate correction for multiple testing

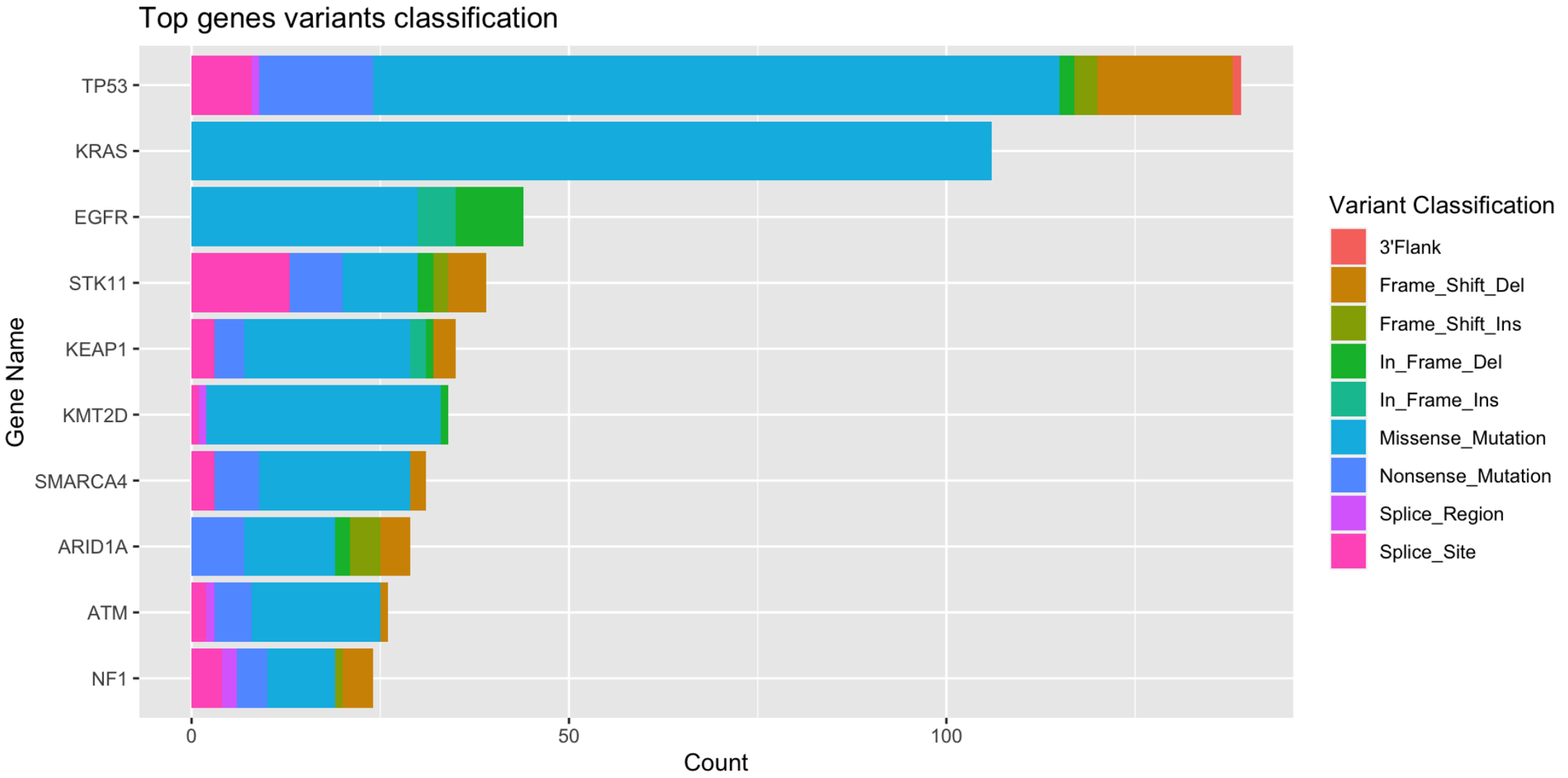
Visualize Data

```
1  gnomeR::ggcomut(nsclc_cohort$cohort_mutations_extended)
```



Visualize Data

```
1  gnomeR::ggtopgenes(nsclc_cohort$cohort_mutations_extended)
```

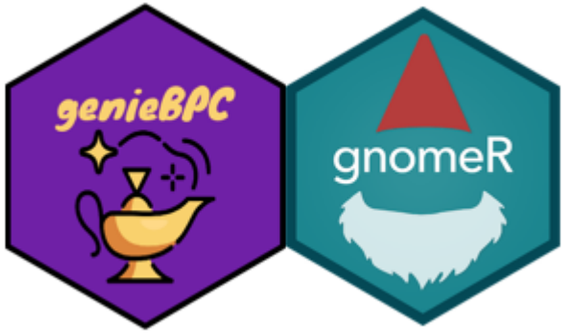


Additional Items & Next Steps

- Additional **visualizations** and **color palettes** useful for genomic data are available in {gnomeR} package
- Some data may require additional data checks. See {gnomeR vignette} for helpful tips on data QA.
- It may be appropriate to **oncoKB annotate** your data and only analyze oncogenic mutations (see oncoKB.org for more information).

Conclusion

- The **{genieBPC}** & **{gnomeR}** R packages offer a reproducible pipeline to create cohorts for clinico-genomics analyses
- **{genieBPC}** streamlines data access and clinical data processing from multiple clinical data files of varying structure to create analytic cohorts
- **{gnomeR}** facilitates annotation and analysis of complicated genomic data.



Note: {gnomeR} can be used for genomic data processing and analyses outside of the GENIE BPC project!

Thank You!

Thanks to all {gnomeR} & {genieBPC} authors & contributors:

Michael Curry, Axel Martin, Arshi Arora, Allie Reiner, Akriti Mishra, Dan Sjoberg, Jessica Flynn, Caroline Kostrzewa, Christine Zhou, Esther Drill, Margie Hannum, and more.