Unified access to cancer proteogenomics data

Caleb M. Lindgren, Hannah Boekweg, David W. Adams, Sadie Tayler, CPTAC Investigators, Samuel H. Payne

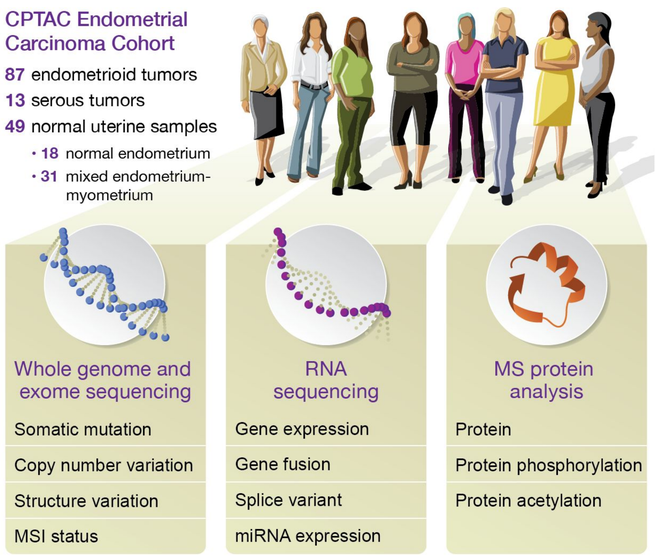
Department of Biology, Brigham Young University, Provo, UT, USA

Abstract

We present a new method for data sharing across large collaborations to improve reproducibility and transparency, by creating a Python package that serves as an interface (API) to the multi-omics characterization of tumors from NCI's CPTAC program.

Introduction

Cancer data has many audiences, including clinicians, biologists, data scientists and patients. Sharing data and analyses across these diverse audiences is challenging. In particular, we want to simplify the link between data and analysis scripts to enable easier data exploration. We embed NCI's CPTAC data into a software API. Each tumor type describes samples with clinical, omics, and imaging data.



Conclusions

The `cptac` Python package brings cancer data to dispersed collaborative groups. Our package incorporates multiple data sets and lowers the entry barrier, expanding our audience while improving reproducibility and transparency.

Results

A Python package, `cptac`, facilitates access to all CPTAC data. Using the package as the single point of data access unifies and simplifies analysis methods across diverse consortia. Current tumor types include: colon, ovarian, endometrial, renal, breast, lung adeno, glioblastoma and head/neck.

Features

- One step installation via pip
- Consistent data and formats across cancers
- Data presented in Pandas dataframes, meaning no need for writing parsers
- API facilitates joining across data types
- Seamlessly works with numeric and graphing libraries (numpy, pandas, matplotlib, seaborn, etc.)
- Versioned data releases
- Package automatically checks that it's up to date, and that the latest version of the data is installed

Loading data tables

```
import cptac
en = cptac.Endometrial()

proteomics = en.get_proteomics()
proteomics.head(3)
```

Sample_ID	A1BG	A2M	A2ML1	A4GALT	AAAS	AACS	AADAT	AAED1	AAGAB	AAK1	...	ZSWIM8	ZSWIM9	ZW10	ZWILCH	ZWINT	ZXDC	ZYG11B
S001	-1.180	-0.863	-0.802	0.222	0.256	0.665	1.2800	-0.3390	0.412	-0.664	...	-0.08770	NaN	0.0229	0.109	NaN	-0.332	-0.4330
S002	-0.685	-1.070	-0.684	0.984	0.135	0.334	1.3000	0.1390	1.330	-0.367	...	-0.03560	NaN	0.3630	1.070	0.737	-0.564	-0.0046
S003	-0.528	-1.320	0.435	NaN	-0.240	1.040	-0.0213	-0.0479	0.419	-0.500	...	0.00112	-0.145	0.0105	-0.116	NaN	0.151	-0.0740

3 rows × 10999 columns

```
clinical = en.get_clinical()
clinical.head(3)
```

Sample_ID	Patient_ID	Proteomics_Tumor_Normal	Country	Histologic_Grade_FIGO	Myometrial_invasion_Specify	Histologic_type	Treatment_naive	Tumor_
S001	C3L-00006	Tumor	United States	FIGO grade 1	under 50 %	Endometrioid	YES	N
S002	C3L-00008	Tumor	United States	FIGO grade 1	under 50 %	Endometrioid	YES	N
S003	C3L-00032	Tumor	United States	FIGO grade 2	under 50 %	Endometrioid	YES	N

Comparing clinical attributes

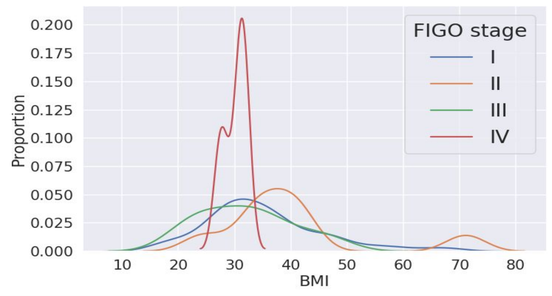
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import cptac

en = cptac.Endometrial()
clinical_data = en.get_clinical()
simplified_clinical = clinical_data[['FIGO_stage', 'BMI']]

pd.set_option('mode.chained_assignment', None)
simplified_clinical.loc[simplified_clinical.FIGO_stage == 'IA', 'FIGO_stage'] = 'I'
simplified_clinical.loc[simplified_clinical.FIGO_stage == 'IB', 'FIGO_stage'] = 'I'
simplified_clinical.loc[simplified_clinical.FIGO_stage == 'IIIA', 'FIGO_stage'] = 'III'
simplified_clinical.loc[simplified_clinical.FIGO_stage == 'IIIB', 'FIGO_stage'] = 'III'
simplified_clinical.loc[simplified_clinical.FIGO_stage == 'IIIC1', 'FIGO_stage'] = 'III'
simplified_clinical.loc[simplified_clinical.FIGO_stage == 'IIIC2', 'FIGO_stage'] = 'III'
simplified_clinical.loc[simplified_clinical.FIGO_stage == 'IVB', 'FIGO_stage'] = 'IV'
figo_stage_bmi = simplified_clinical.sort_values('FIGO_stage')

sns.set(style='darkgrid', context="talk", font_scale=1.25)
NUM_COLORS = 12

figoHist = sns.FacetGrid(figo_stage_bmi, hue="FIGO_stage", legend_out=False, height=7, aspect=1.5)
figoHist = figoHist.map(sns.kdeplot, "BMI")
figoHist.set(ylabel='Proportion')
plt.legend(title="FIGO_stage", fontsize='28',title_fontsize='28')
plt.savefig("figo_and_bmi.png")
```



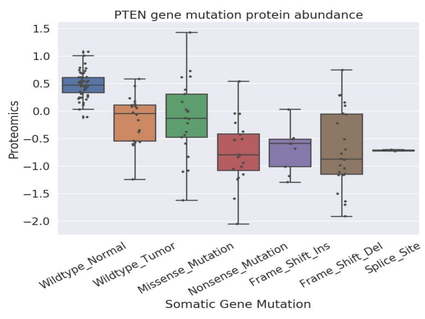
Mutation effects on the proteome

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import cptac
en = cptac.Endometrial()

somatic_mutations = en.get_mutations()
gene = somatic_mutations["Gene"].value_counts().index[0]
omics_mutations = en.join_omics_to_mutations(omics_df_name="proteomics", mutations_genes=gene,
                                             omics_genes=gene, mutations_filter=[])

plt.figure(figsize=(11.7, 8.27))
sns.set(style='darkgrid', context="talk", font_scale=1.25)
somatic_boxplot = sns.boxplot(data=omics_mutations, x=gene + "_Mutation", y=gene + "_proteomics",
    order=["Wildtype_Normal", "Wildtype_Tumor", "Missense_Mutation", "Nonsense_Mutation",
           "Frame_Shift_Ins","Frame_Shift_Del","Splice_Site"])
somatic_boxplot.set_title("PTEN gene mutation protein abundance")
somatic_boxplot = sns.stripplot(data=omics_mutations, x=gene + "_Mutation", y=gene + "_proteomics",
                                jitter=True, color=".3",
                                order=["Wildtype_Normal", "Wildtype_Tumor", "Missense_Mutation",
                                       "Nonsense_Mutation", "Frame_Shift_Ins", "Frame_Shift_Del",
                                       "Splice_Site"])
somatic_boxplot.set(xlabel="Somatic Gene Mutation",ylabel="Proteomics")
somatic_boxplot.set_xticklabels(somatic_boxplot.get_xticklabels(),rotation=30)
plt.savefig("prot_and_mut.png", bbox_inches='tight')
```



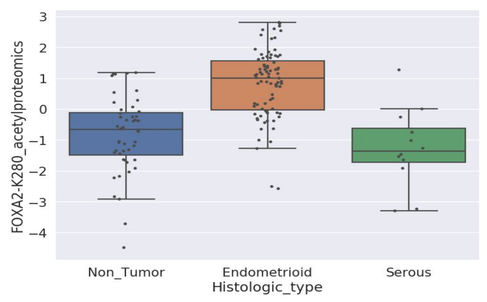
Comparing histology with acetylation

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import cptac

en = cptac.Endometrial()
clinical_attribute = "Histologic_type" # Set desired attribute to variable 'clinical_attribute'
clinical_and_acetylation = en.join_metadata_to_omics( # Merge attribute with acetylation dataframe
    metadata_df_name="clinical",
    omics_df_name="acetylproteomics",
    metadata_cols=clinical_attribute)

# Make dataframes with only endometrioid and only serous data in order to compare
endom = clinical_and_acetylation.loc[clinical_and_acetylation[clinical_attribute] == "Endometrioid"]
serous = clinical_and_acetylation.loc[clinical_and_acetylation[clinical_attribute] == "Serous"]
clinical_and_acetylation[[clinical_attribute]] = clinical_and_acetylation[[clinical_attribute]].fillna(
    value="Non_Tumor") # Set the NaN values to "Non_Tumor"

graphingSite = 'FOXA2-K280 acetylproteomics'
plt.figure(figsize=(11.7, 8.27))
sns.set(style='darkgrid', context="talk", font_scale=1.25)
sns.boxplot(
    x=clinical_attribute,
    y=graphingSite,
    data=clinical_and_acetylation,
    showfliers=False,
    order=["Non_Tumor", "Endometrioid", "Serous"])
sns.stripplot(
    x=clinical_attribute,
    y=graphingSite,
    data=clinical_and_acetylation,
    color=".3", # Makes dots black
    order=["Non_Tumor", "Endometrioid", "Serous"])
plt.savefig("acet_and_hist.png")
```



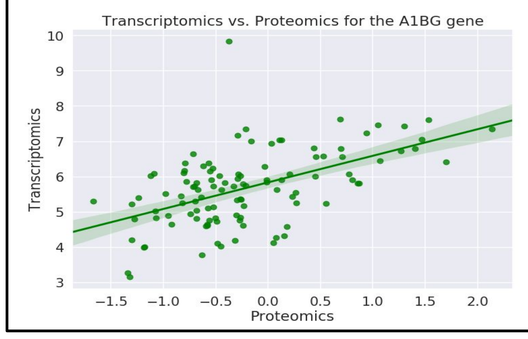
Comparing multi-omics data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import cptac
en = cptac.Endometrial()

A1BG_cross = en.join_omics_to_omics(df1_name="proteomics", df2_name="transcriptomics",
                                    genes1="A1BG", genes2="A1BG")

plt.figure(figsize=(11.7, 8.27))
sns.set(style="darkgrid", context="talk", font_scale=1.25)
plot = sns.regplot(x=A1BG_cross.columns[0], y=A1BG_cross.columns[1], data=A1BG_cross, color="green")
plot.set(xlabel='Proteomics', ylabel='Transcriptomics',
         title='Transcriptomics vs. Proteomics for the A1BG gene')
plt.savefig("prot_and_tran.png")
```



Acknowledgments: National Cancer Institute (NCI) CPTAC award U24 CA210972.

Contact: calebmlindgren@gmail.com; hannahboekweg@gmail.com; sam_payne@byu.edu

https://payne.byu.edu github.com/PayneLab @byu_sam