Protocol capture for running repertoire PCA analysis

Introduction

The following is a protocol capture of how to run repertoire PCA analysis according to Sevy, Soto *et al.* 2019. All the Python scripts necessary to run the analysis are present in this directory.

Note that all analysis scripts will only function properly if they are in the correct directory as provided.

Dependencies

These scripts require the use of Python 2.7 as well as the scikit-learn package (http://scikit-learn.org/stable/index.html), tqdm (https://github.com/tqdm/tqdm), pandas (https://pandas.pydata.org/), numpy (http://www.numpy.org/), matplotlib (https://matplotlib.org/), and seaborn (https://seaborn.pydata.org/). We recommend installing all necessary packages before beginning this protocol.

Overview of steps

First I will quickly review the steps to run repertoire PCA analysis. Your input files should be tab-separated value (tsv) files which are tables of IGHV-IGHJ pairs. These tables should be raw counts, not frequencies – the frequencies will be calculated during the protocol. You will begin by creating subsampled replicates of your dataset to subsample to a common depth. This is a crucial part of the protocol that allows you to compare datasets of different sequencing depths. You can control the depth that you want to subsample to (default 10^5) and the number of replicates to make (default 10). After creating the subsampled datasets the data will be normalized by Z score and the PCA will be run on all input datasets. The script will then output several plots summarizing the results.

Subsampling datasets

The first step as previously mentioned is to subsample the datasets. Take a look at an input file, such as hip1.tsv, to see how the data should be formatted:

```
hip1.tsv:

     IGHJ1   IGHJ2   IGHJ3   IGHJ4   IGHJ5   IGHJ6
IGHV1-2 4822    3526    34603   162974  73634   148530
IGHV1-3 2495    1680    13429   75330   32939   56010
IGHV1-8 1622    1603    11868   55952   34357   65835
IGHV1-18    6818    4626    43174   224133  91387   174045
.....
```

This table is the raw count of each IGHV-IGHJ pair. **This data is computed not from the raw sequences, but after grouping the sequences into clonotypes and de-duplicating.** For more detail on the pre-processing steps to get this data see the Methods section in the publication. Note that all V-J pairs in your input table may not be considered during PCA. An initial processing step in the subsampling is eliminating all but a predefined set of 306 V-J pairs, which are defined in the publication.

After gathering your dataset you are ready to run the subsampling step. The following script will do the subsampling:

```
python subsample_datasets.py --reps=10 --depth=100000 --n_jobs=8 hip1.tsv
hip2.tsv hip3.tsv
```

In this example you will only run the analysis on three datasets, the three healthy donors used in this study. However all the tables analyzed in the paper are also in this example folder. The `reps` option controls how many subsampled replicates to make, `depth` controls what sequencing depth to subsample to, and `n_jobs` will run the subsampling on multiple processors to speed up the processing. The output of this script is a folder called “datasets” that contains all the replicate tsv files, called by the name of the input file plus a suffix (*i.e.* hip1_1.tsv, hip1_2.tsv, etc.).

Principal component analysis

Now that the subsampled datasets are prepared you can run the PCA. Use the following script to run PCA and analysis:

```
python analyze_pca.py --output pca --ncomp 8 --reps 10 --path datasets/
hip1 hip2 hip3
```

The key parameters in this script are:

- output: what is the prefix to give the PDFs showing the results? Defaults to pca, so the output PDFs will be pca_explained_variance.pdf, pca_2D_projection.pdf, pca_PC1_heatmap.pdf, and pca_PC2_heatmap.pdf
- ncomp: how many components to keep during the PCA? Defaults to 8. Note that this will show the contribution of the top 8 components, but the 2D projection will still only plot the top 2 dimensions regardless of the value of ncomp.
- reps: how many replicates did you make of the subsampled datasets? Defaults to 10. This tells the script where to find the input files.
- path: where were the replicate datasets saved? Again tells the script where to find input files
- Final positional arguments are hip1 hip2 hip3 in this example – these are the prefixes of the datasets you want to analyze.

This will generate several output figures summarizing the results:

- pca_explained_variance.pdf: a scree plot showing the variance explained by each component. Only shows the number of components defined by ncomp.
- pca_2D_projection.pdf: the values of components 1 and 2 plotted against each other. Note that regardless of the value of ncomp this plot will only show two dimensions.
- pca_PC1_heatmap.pdf: the weight of each feature comprising component 1 shown as a heatmap.
- pca_PC2_heatmap.pdf: the weight of each feature comprising component 2 shown as a heatmap.