CLL scRNA-seq

Brad Blaser

4/24/2020

Introduction

12 patient samples were run using the 10X genomics 5' Immune Profiling kit for the purpose of studying immune modulation resulting from emerging BTK inhibitor resistance.

Sequencing data (10X pipestance folders only) are available in the file paths indicated in "01_load_10X_gex_data.R"

Quality control data still need to be organized/centralized but if memory serves, after rerunning sample 8 it all looked ok.

Raw sequencing data was processed using CellRanger v3.1.0 and analyzed in Monocle v3.

Nanopore sequencing was run and processed by Esko and Thomas. According to Esko:

> The data was:
>
> 1. Base called with Guppy using the high-accuracy model
> 2. Demultiplexed using the min_score 70 cutoff to reduce false positive barcode assignments
> 3. Mapped to GRCh38 with minimap2
> 4. Locus-spanning reads were pulled out and extracted into those separate FASTQs

Thomas processed the data as follows:

> We are using a semi-global alignment that doesn't penalize gaps on either end of the target sequence (the read that we are searching). Our query sequence is : CTACACGACGCTCTTC-CGATCTNNNNNNNNNNNNNNNNNNNNNNNNNNTTTCTTATATGGG
>
> We use a scoring matrix of form:

X| G| A| T| C| N

G| 2|-2|-2|-2| 2

A|-2| 2|-2|-2| 2

T|-2|-2| 2|-2| 2

C|-2|-2|-2| 2| 2

N| 2| 2| 2| 2| 2

> So N matches everything and +2 score for a match and -2 for a mismatch. And we require a score of 85 which is a roughly matching at least 43 out of 61 nucleotides. 26 of the bases will always match so we will have a minimum score of 52. We use a gap open and extension penalty of -2 as well. If the score is greater than 85, we extract the aligned sequence from the read as is with no error correction. This extracted sequence is stored as a bam tag. I use another program to extract a nucleotide at a position of my choosing from the read and that is combined with the bam tag. For example, we are extracting a positive strand nucleotide at mapped position of 101356176 (zero-based) which is the site of the canonical BTK C481S mutation.

Nanopore data is located at X/Labs/Blaser/single_cell/cll_project.

Sample metadata is being collected.

Current R session data is mirrored at

X/Labs/Blaser/single_cell/cll_project/R_session_data/cll_scrnaseq_mirror.RData.

The original CDS elements, post-dimension reduction but without all the other stuff is in cll_original_cds_elements.RData. Start from here if you want a clean slate but still want the same UMAP coordinates.

Monocle analysis

52373 cells were captured in 12 samples (4 patients x 3 timepoints):

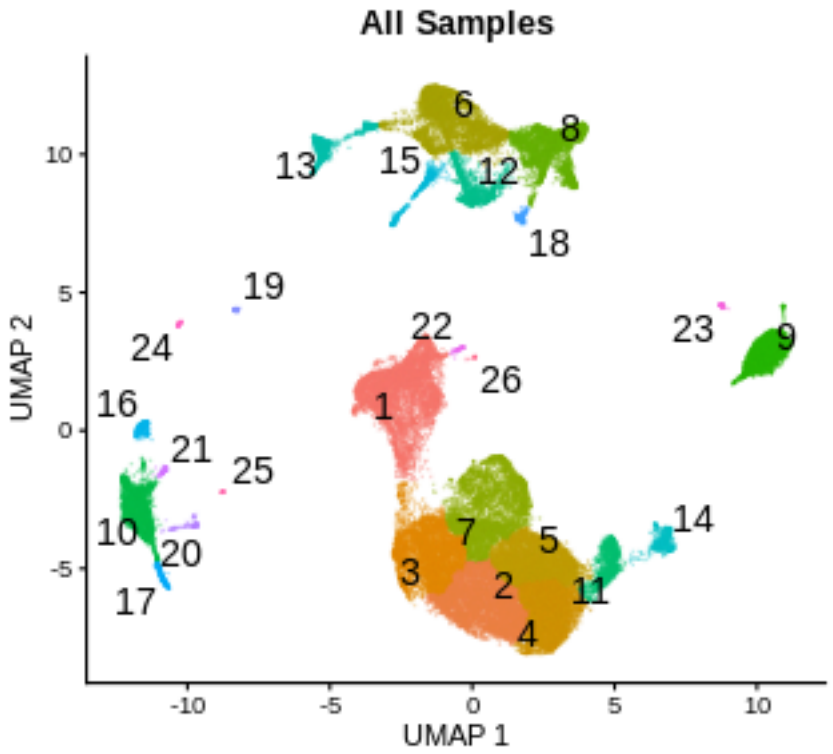
```
print(top_level_summary)
```

```
## # A tibble: 12 x 3
##    pt_timepoint   total_cells running_total
##    <chr>                <int>         <int>
##  1 cll5_baseline         3731          3731
##  2 cll5_btk_clone        6769         10500
##  3 cll5_relapse          6349         16849
##  4 cll6_baseline        10794         27643
##  5 cll6_btk_clone        3528         31171
##  6 cll6_relapse          2429         33600
##  7 cll7_baseline         3868         37468
##  8 cll7_btk_clone        2693         40161
##  9 cll7_relapse          2907         43068
## 10 cll8_baseline         3800         46868
## 11 cll8_btk_clone        2120         48988
## 12 cll8_relapse          3385         52373
```

Ribosomal genes were removed, and umap coordinates were calculated using batch-corrected principle components. Batch correction was performed using the patient variable.

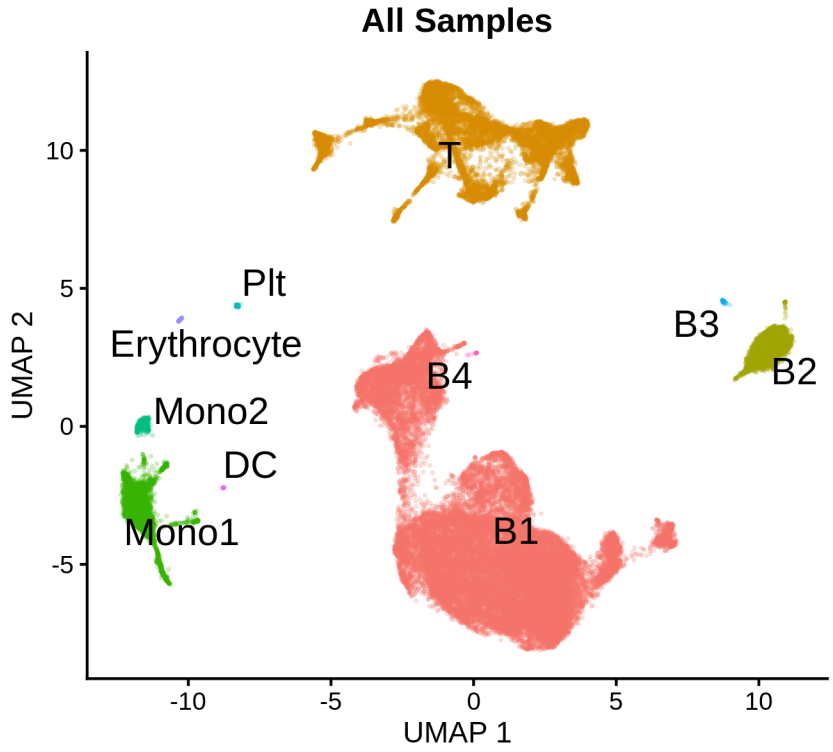
Cells were clustered using the "Leiden" method with the default setting of k = 20.

```
plot(leiden_cluster_plot_all)
```



This is too many clusters for now. We can come back to it. Partitions are easier to work with. 10 were assigned according to top markers identified using Jensen-Shannon Specificity and logistic regression:

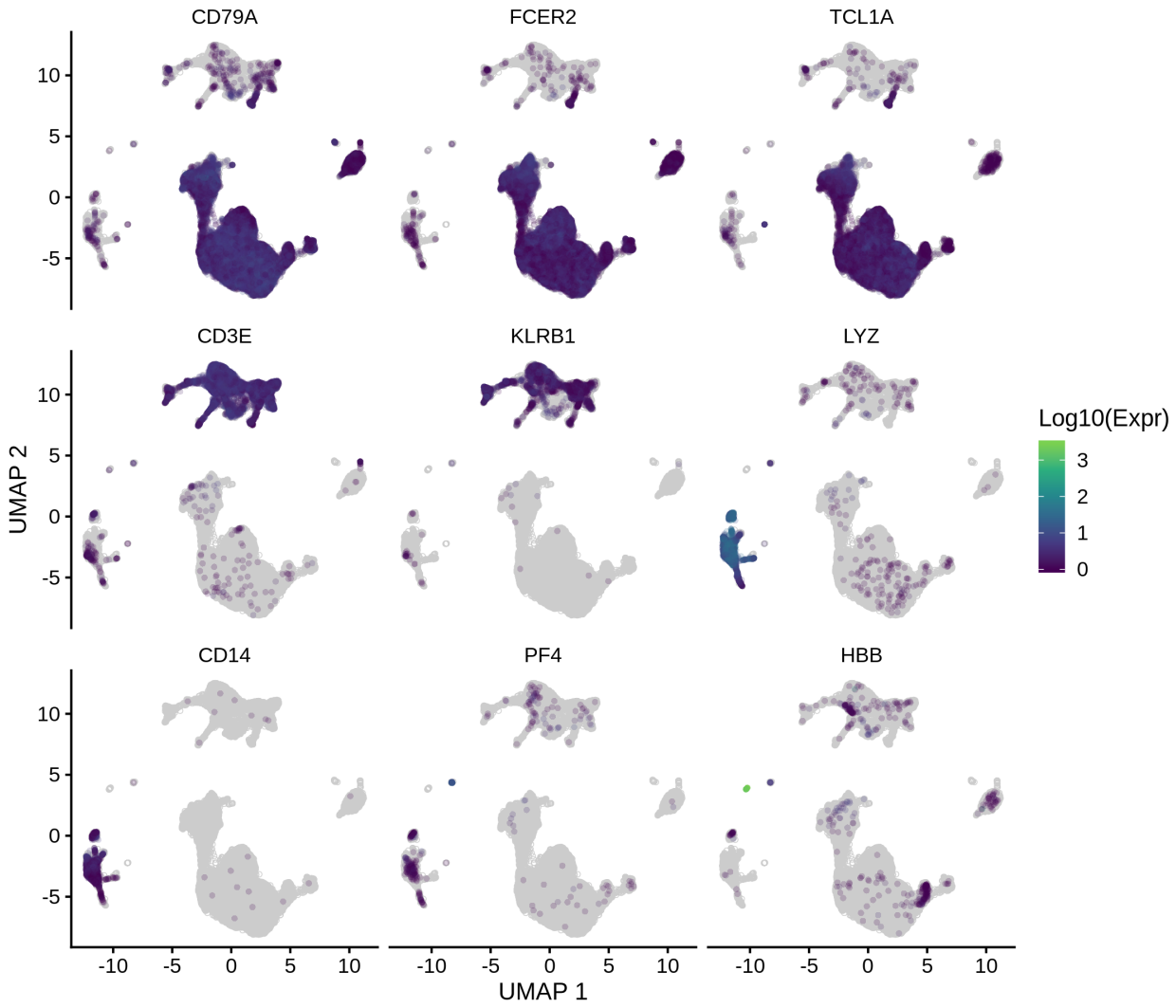
```
plot(cluster_plot_all)
```



Top cluster and partition markers can be found in data_out/.

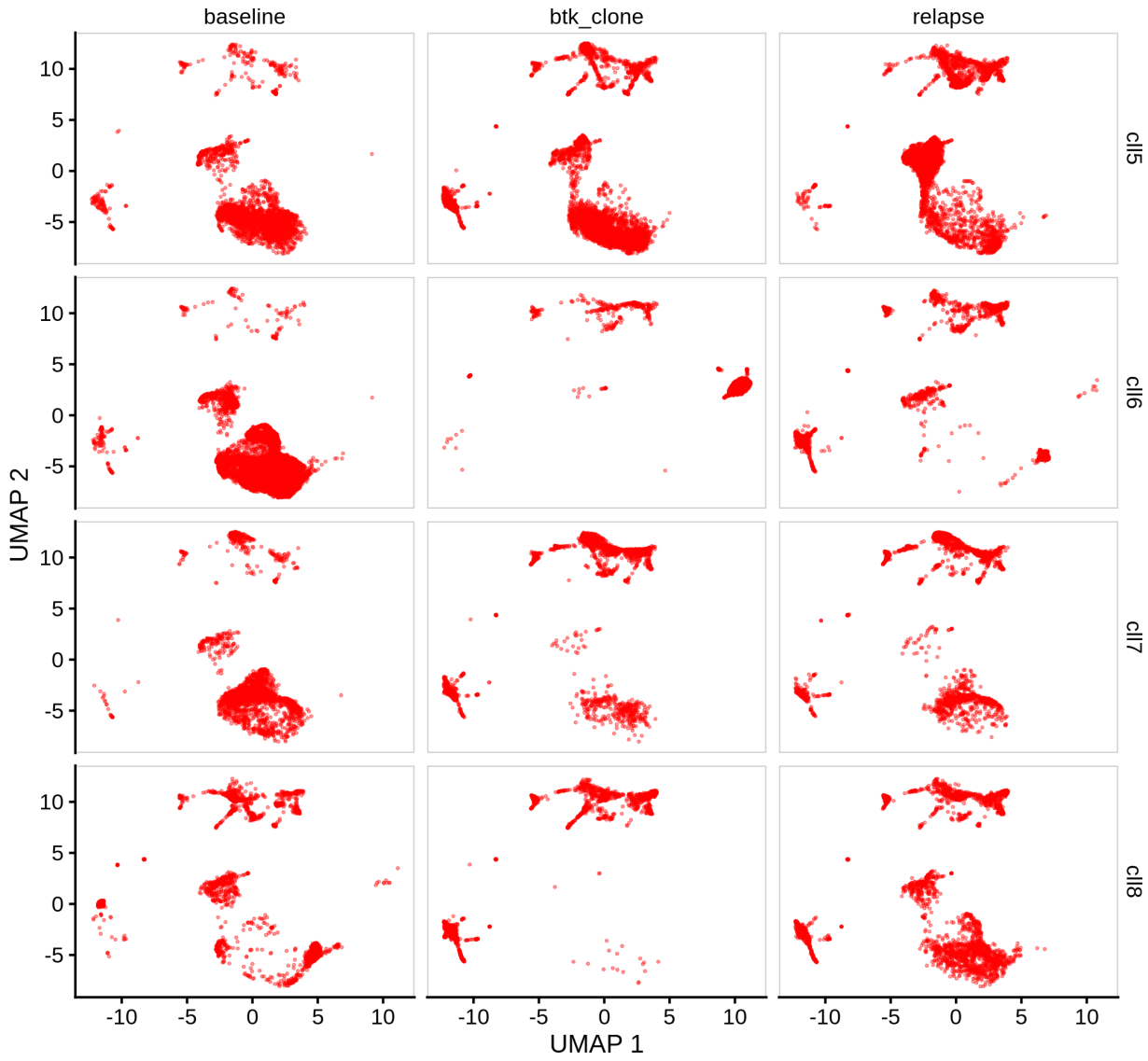
UMAP plots showing representative markers:

```
plot(marker_genes)
```



UMAPs plotted by sample:

```
plot(pt_timepoint_facet_plot)
```



My high-level impression of these plots is that there are no major differences in the non-B cell populations, allowing for cell numbers. CLL6 was an outlier in terms of the B cell population at the btk_clone timepoint. Maybe that will turn out to be interesting. The other patients' B cells largely stayed within the large B1 cluster, possibly with some variation within. This is where we can use the Leiden clusters.

BTK Genotyping

Fastqs from two nanopore runs were provided. The first was pts 5-7 and the second was pts 5-8. Fastqs are in the form of:

```
@0_v1

ANNNACCCACGACGCTCTTCCGATCTAGATCTGCACGACGAACTCCGTTCTTTTTCTTATATGGG

+
```

```
5555555555555555555555555555555555555555555555555555555555555555
```

The first nucleotide represents the BTK hotspot nucleotide. C = WT, A = mutant.

WT and mutant reads were selected by filtering for those starting with C and A, respectively, using Cutadapt. All other reads were discarded.

Next, FASTA files were made in the following form:

```
>AAACCTGAGAGTAATC-1_cll5_baseline

CTACACGACGCTCTTCCGATCTAAACCTGAGAGTAATC
```

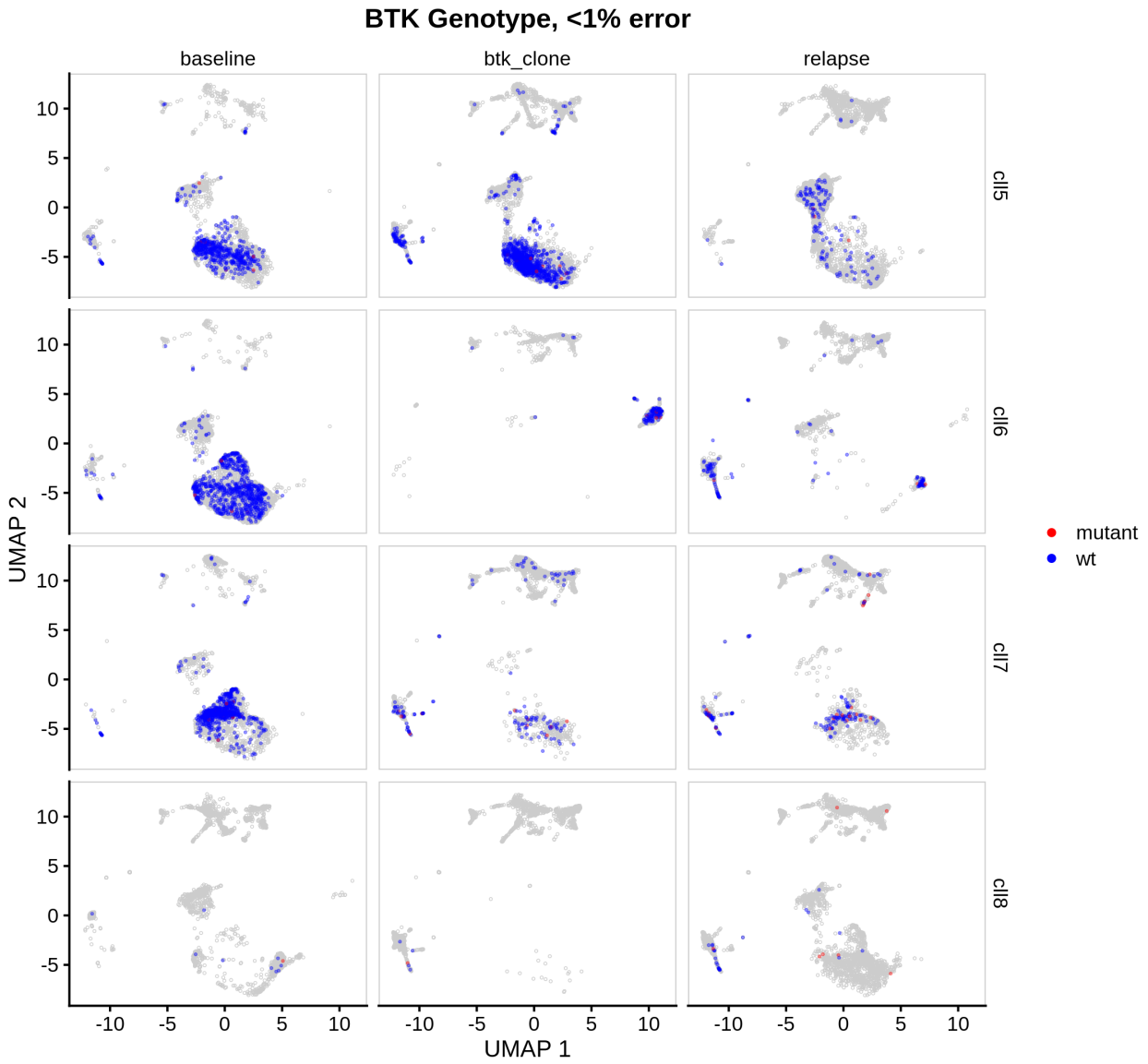
The sequence name is a composite of the 10X cell barcode, the patient and the timepoint. The sequence is contains the partial illumina adapter followed by the 10X barcode for all cells in the dataset. These fastas are input as adapter sequences to Cutadapt which is used to match genotyped reads to cell barcodes.

Cell barcode matching was performed with two degrees of error tolerance. The error rate is defined as the number of mismatched bases x the adapter length. Mismatches and indels are both considered. All adapters are 38 nt long so an error tolerance of 10% would permit 3 errors in the adapter string but not 4. An error tolerance of 1% would permit no errors. The analysis was run with error tolerance equal to 10% and 1%.

If any cell expressed a mutant read it was called mutant. 1 read assigned to a cell barcode was considered sufficient to provide a genotype.

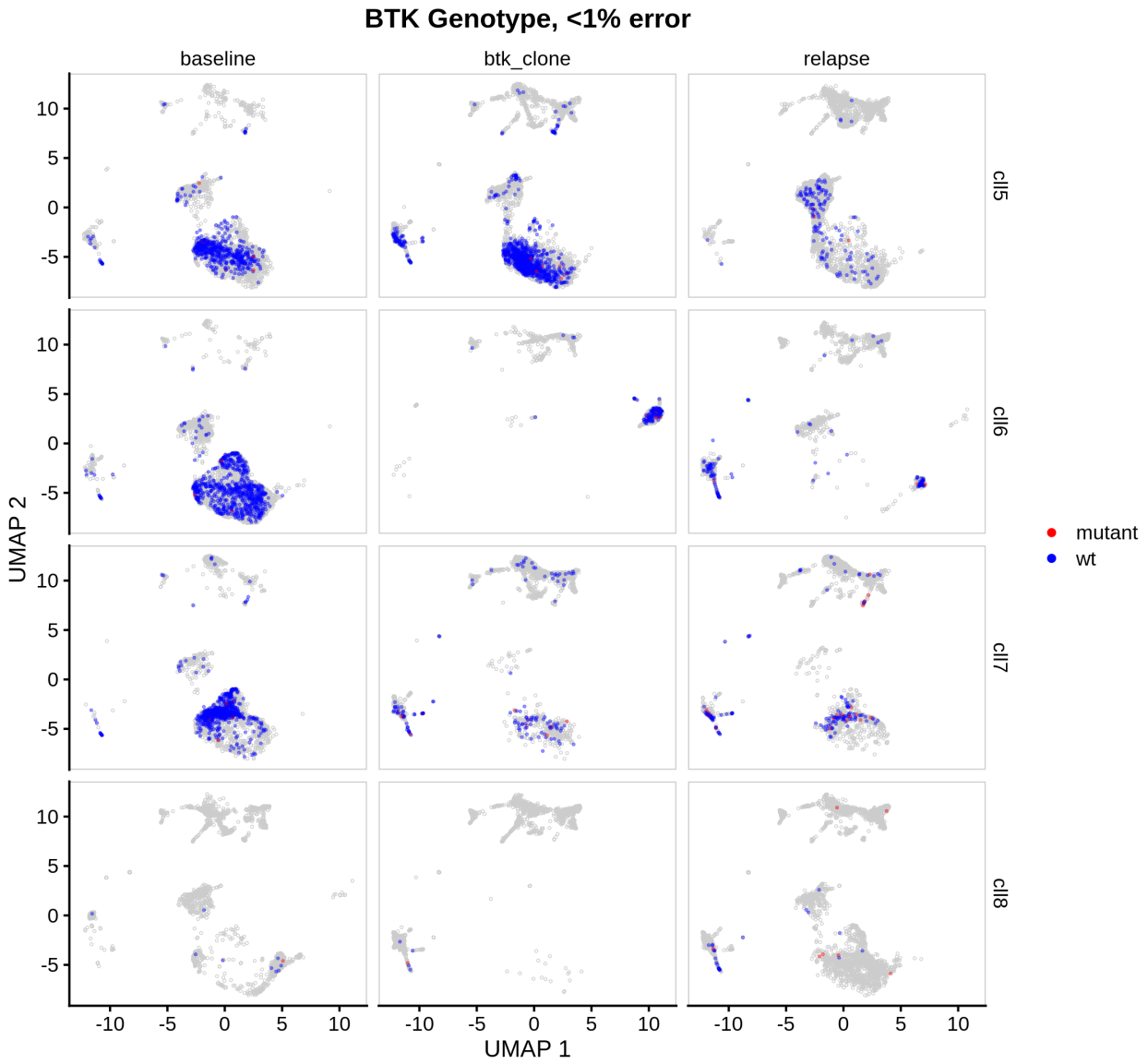
Here is the plot for BTK genotypes using an error threshold of 1%:

```
plot(btk_e1pct_facet)
```



And 10% error threshold:

```
plot(btk_e1pct_facet)
```

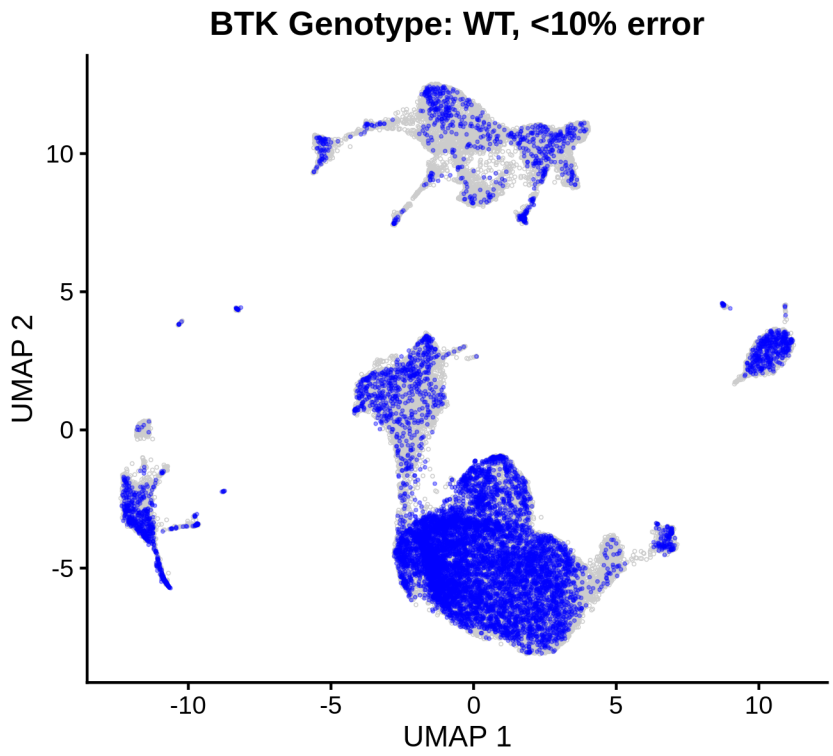


To me they don't look markedly different, other than we are calling more cells with 10% error. I had previously done extensive analysis looking at the possibility of barcode misassignment. It doesn't look like the barcodes are really being misassigned, even with 10% threshold. We have no way of correcting the hotspot nucleotide call, so if we want to track down the mutant BTK reads inside myeloid cells we will have to use Pac Bio or some orthogonal approach (or both).

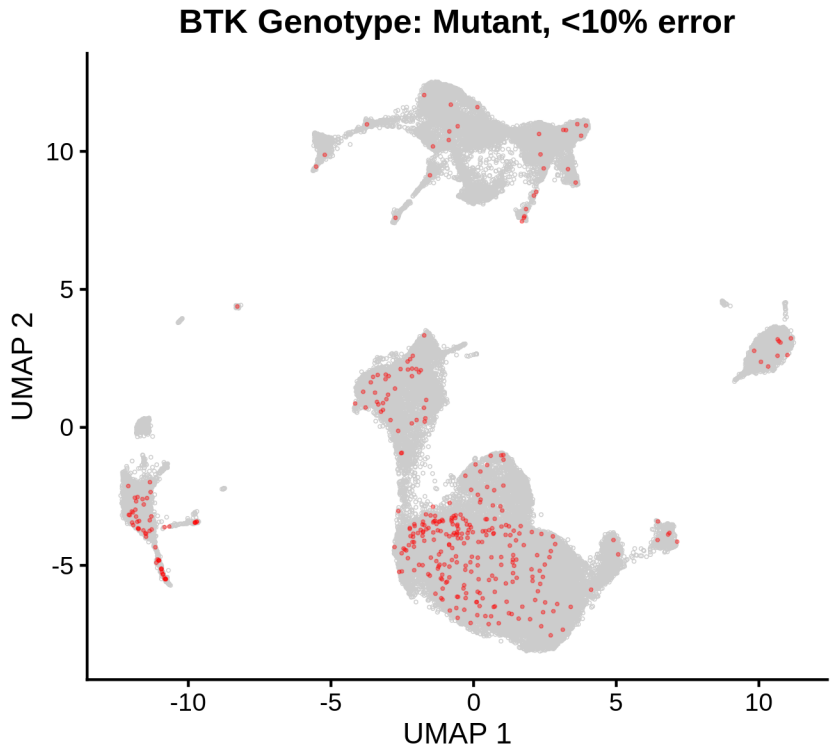
I overlaid all of the BTK genotyping reads to see if they consistently clustered in a specific region or not:

```
btk_overlays
```

```
## [[1]]
```

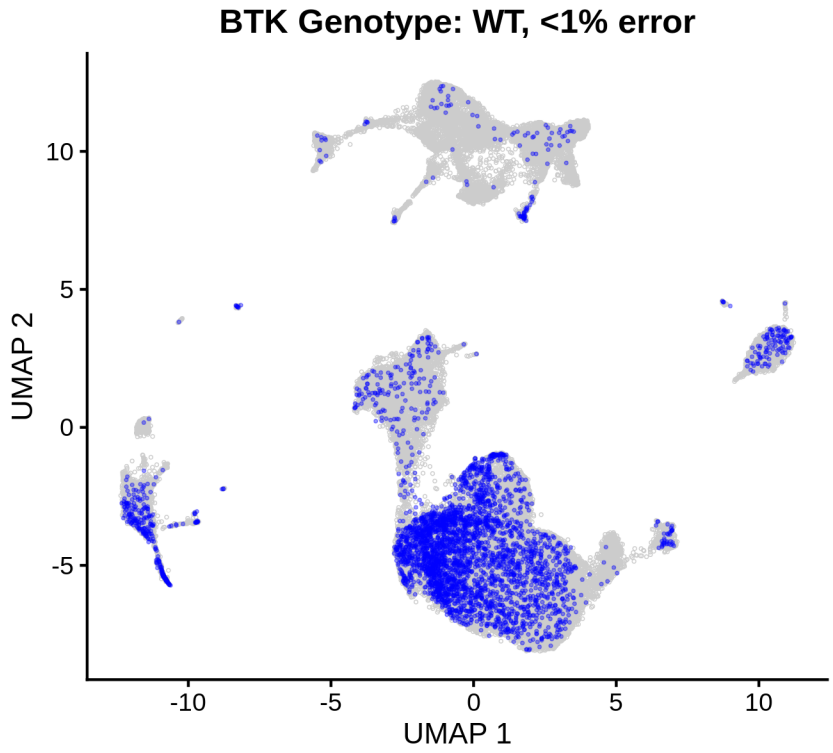

BTK Genotype: WT, <10% error
UMAP 2
UMAP 1
10
5
0
-5
-10
-5
0
5
10

##
[[2]]

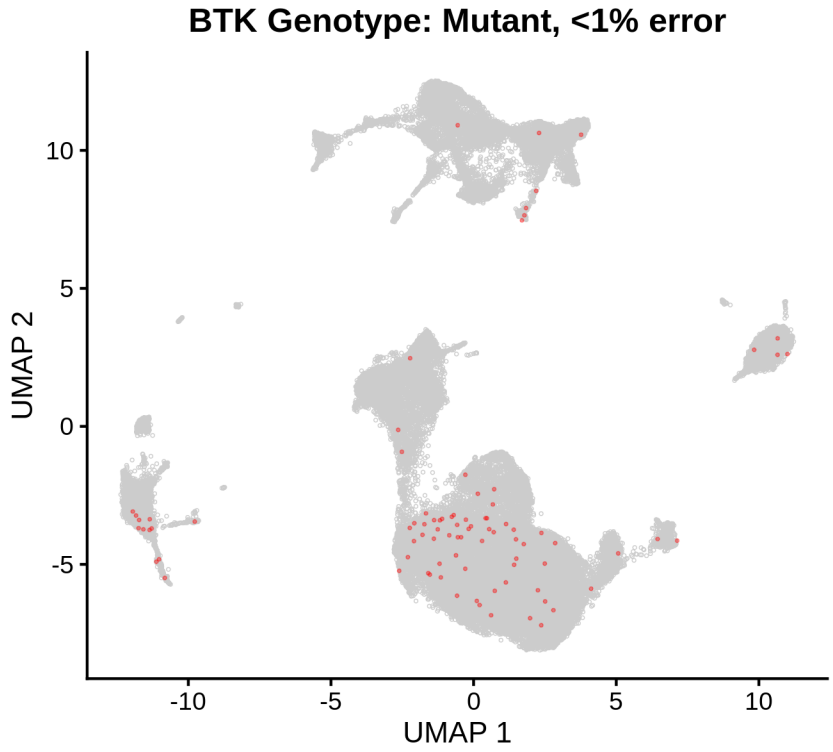

BTK Genotype: Mutant, <10% error
UMAP 2
UMAP 1
10
5
0
-5
-10
-5
0
5
10

##

```
## [[3]]
```


BTK Genotype: WT, <1% error
UMAP 2
UMAP 1
10
5
0
-5
-10
-5
0
5
10

```
##
## [[4]]
```

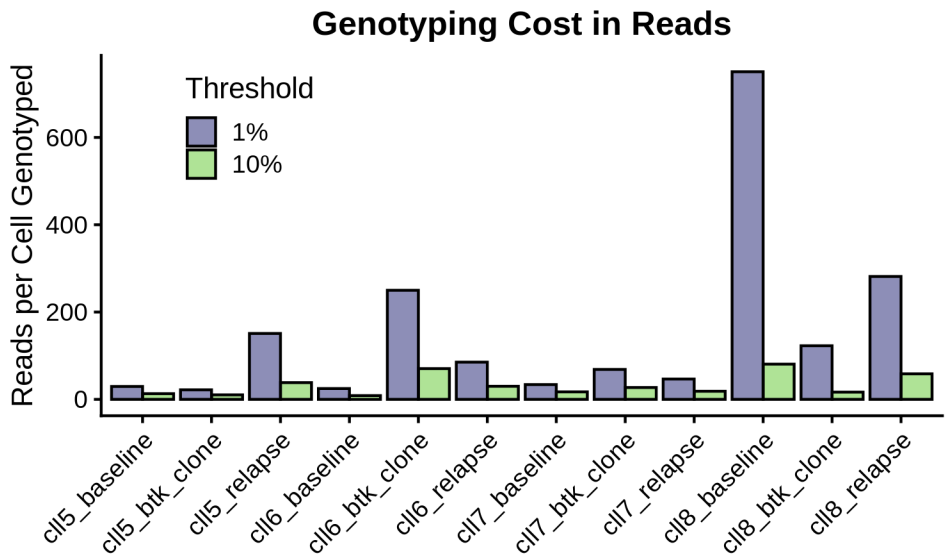

BTK Genotype: Mutant, <1% error
UMAP 2
UMAP 1
10
5
0
-5
-10
-5
0
5
10

To me it seems the answer is not so much.

Stats for the BTK genotyping are here: data_out/btk_stats.csv. This includes the number of cells per sample per partition, the number genotyped, and the mean error in BTK cell barcode assignment of genotyped cells in that group. Generally the mean error at 10% threshold is between 2 and 3 over the 38 nt stretch.

I calculated the "Genotyping Cost" in terms of BTK reads per cell genotyped at the 10% and 1% thresholds. It looks like patient 8 had a lot more unusable reads at the 1% threshold. Not sure how our sequencing approach changed for that one.

```
plot(genotyping_cost)
```




```
print(cell_genotype_efficiency,width = Inf)
```

```
## # A tibble: 12 x 7
##    pt_timepoint   total_cells btk_cells_1 btk_cells_10 read_count
##    <chr>                <int>       <int>        <int>      <dbl>
##  1 cll5_baseline         3731         521         1179      15389
##  2 cll5_btk_clone        6769        1061         2252      23199
##  3 cll5_relapse          6349         139          547      20984
##  4 cll6_baseline        10794         731         2105      18114
##  5 cll6_btk_clone        3528         117          415      29236
##  6 cll6_relapse          2429         109          310       9295
##  7 cll7_baseline         3868         761         1500      25761
##  8 cll7_btk_clone        2693         137          346       9393
##  9 cll7_relapse          2907         156          391       7265
## 10 cll8_baseline         3800          10           93       7505
## 11 cll8_btk_clone        2120           5           37        614
## 12 cll8_relapse          3385          32          154       9008
##    reads_per_cell_1 reads_per_cell_10
##               <dbl>             <dbl>
##  1             29.5              13.1
##  2             21.9              10.3
##  3            151.               38.4
##  4             24.8               8.61
```

```
##  5            250.             70.4
##  6             85.3            30.0
##  7             33.9            17.2
##  8             68.6            27.1
##  9             46.6            18.6
## 10            750.             80.7
## 11            123.             16.6
## 12            282.             58.5
```

Conclusion

I think that the btk genotyping is still a novelty although it becomes less so as time passes. As long as we don't try to drill down and compare really small numbers of cells, I think we can feel comfortable that the cell barcodes are being assigned correctly. It just doesn't seem that the error threshold of 10% looks much different from 1% (which is a perfect match), except that you have more cells.

We need to think about confirmatory approaches for the mutant reads assigned to non-B cells. Understanding why these are there could be interesting. Phagocytosis? Convergent evolution? But the fact that we see mutant reads in the dx timepoint suggests we have some background error to deal with.

I'm not sure we are going to find anything inherently different about the mutant and WT BTK cells. One potentially interesting thing is why pt 6 at btk_clone timepoint had such a different profile from the others. After accounting for patient:patient variation, the rest of the CLL cells just sit there. This isn't what we cell with AML.

I think if we had another 4 patients, this would be a reasonable dataset to publish, even as a descriptive paper. We are already at 50k cells. Seems like we have some funds left on the account. We still need to get the clinical metadata in to see if there are patients we want to select or avoid in the next 4.

Let me know what you think.