**Data location:**

1. **Server Info:**

Name: Markey Cancer Center Cancer Research Informatics (CRI) HPC

Login (Use UK account):

1. Use the Global Protect client to connect to VPN at ra.mccp.uky.edu
2. Terminal connection: ssh UKACCOUNTNUM4@hpccontrol.kcr.uky.edu
3. FileZilla connection:
   1. Host: hpccontrol.kcr.uky.edu
   2. Protocol: SFTP – SSH File Transfer Protocol
4. **Local directory info:**

ReCAPSE\_Project.zip

1. **Raw data and dictionary location:** 
   1. Server:
      1. /recapse/data/Testing data for UH3 - Dec 16 2020/
      2. /recapse/data/Code\_Groups/
   2. Local:
      1. ReCAPSE\_Project / ReCAPSE\_Data /Testing data for UH3 - Dec 16 2020/
      2. ReCAPSE\_Project /ReCAPSE\_Data/Code\_Groups/
2. **Intermediate output data location:**
3. Server: /recapse/intermediate\_data/
4. Local: ReCAPSE\_Project /ReCAPSE\_Intermediate\_Data/0610\_21/
5. **Most recent performance’s location:**
   1. Server:

/recapse/intermediate\_data/16\_Performance\_WithSurgPrimSite\_V1\_1217updated/

* 1. Local: ReCAPSE\_Project/ReCAPSE\_Intermediate\_Data/0610\_21/16\_Performance\_WithSurgPrimSite\_V1\_1217updated/

1. **Slides:**

Local: ReCAPSE\_Project /ReCAPSE\_Slides/

1. **FredHutch materials:**

Local: ReCAPSE\_Project /UH3 cancer recurrence/

**Steps:**

1. **Extract raw unique codes** 
   * **Code:** 0A\_ReadLargeFileAndGetUniqueCodes.R
   * **Code description:** Extract raw unique codes from Medicare and Medicaid claims data.
   * **Input files:** 
     1. kcr\_medicaid\_healthclaims\_fb0015.csv
     2. KCR\_MEDICAID\_PHARMCLAIMS\_FB0015.csv
     3. kcr\_medicare\_claims\_fb0015.csv
   * **Output:**
     1. Output dir: 0\_Codes/BeforeClean\_UniqueCodes/
     2. Output files:
        1. 0\_unique\_Diag\_Codes\_Medicaid.xlsx
        2. 0\_unique\_Proc\_Codes\_Medicaid.xlsx
        3. 0\_unique\_Drug\_Codes\_Medicaid.xlsx
        4. 0\_unique\_Diag\_Codes\_Medicare.xlsx
        5. 0\_unique\_Drug\_Codes\_Medicare.xlsx
        6. 0\_unique\_Proc\_Codes\_Medicare.xlsx
     3. Output example:

|  |  |  |
| --- | --- | --- |
| CODE | TYPE | CLAIM |
| CODE\_40391 | DIAG\_ICD9or10 | Medicaid |
| CODE\_585 | DIAG\_ICD9or10 | Medicaid |
| CODE\_71516 | DIAG\_ICD9or10 | Medicaid |

1. **Clean unique codes in output files in I.**
   * **Code:** 0B\_CleanUniqueCodes.R
   * **Code description:**
     1. replace any codes with non-alphanumeric characters with ""
     2. replace decimal with ""
     3. replace space with ""
     4. reformat code based on code type.
        1. For HCPC, if less than 5 char long; for ICD, if less than 3 char long, do the following:
           1. if numeric, prepending "0"s to match the code minimum length
           2. if non-numeric, remove
        2. if DRUG\_NDC or DRUG\_THERA\_CLS\_AHFS, remove leading 0s
     5. Remove the unspecified codes.
        1. e codes
        2. External cause codes
     6. Remove duplicates and NAs after reformatting
     7. Add Drug names, GNN and short GNN to each drug codes
   * **Input files:** 
     1. 0\_unique\_Diag\_Codes\_Medicaid.xlsx
     2. 0\_unique\_Proc\_Codes\_Medicaid.xlsx
     3. 0\_unique\_Drug\_Codes\_Medicaid.xlsx
     4. 0\_unique\_Diag\_Codes\_Medicare.xlsx
     5. 0\_unique\_Drug\_Codes\_Medicare.xlsx
     6. 0\_unique\_Proc\_Codes\_Medicare.xlsx
     7. DrugList.csv
   * **Output:**
     1. Output dir: 0\_Codes/AfterClean\_UniqueCodes/
     2. Output files:
        1. 0\_Cleaned\_Unique\_Diag\_Codes.xlsx
        2. 0\_Cleaned\_Unique\_Proc\_Codes.xlsx
        3. 0\_Cleaned\_Unique\_Drug\_Codes.xlsx
     3. Output example:

|  |  |  |
| --- | --- | --- |
| CODE | TYPE | CLAIM |
| 40391 | DIAG\_ICD9or10 | BOTH |
| 585 | DIAG\_ICD9or10 | BOTH |
| 71516 | DIAG\_ICD9or10 | BOTH |

1. **Group cleaned unique codes** 
   * **Code:** 0C\_Code\_Grouping.R
   * **Code description:** 
     1. Group diagnosis and procedures codes into CCS, Chubak, and Ritzwoller groups and add description of Codes
     2. Group drug codes into DM3 groups using short GNN of drug codes and add description of codes
   * **Input files:** 
     1. Grouping file:
        1. icd9\_dxref 2015.csv
        2. ccs\_dx\_icd10cm\_2019\_1.csv
        3. icd9\_prref 2015.csv
        4. ccs\_pr\_icd10pcs\_2020\_1.csv
        5. CCS\_services\_procedures\_v2021-1.csv
        6. BRAVA\_lookup.20180502.edit.csv
        7. Ritzwoller\_code\_table.edited.csv
        8. Drug Code Groups-DM3.sorted.csv
        9. Val\_Quan\_Final SecondRoot List and NDC.xlsx
     2. Unique codes files:
        1. 0\_Cleaned\_Unique\_Diag\_Codes.xlsx
        2. 0\_Cleaned\_Unique\_Proc\_Codes.xlsx
        3. 0\_Cleaned\_Unique\_Drug\_Codes.xlsx
   * **Output:**
     1. Output dir: /0\_Codes/Grouped\_CleanUniqueCodes/
     2. Output files:
        1. Unique\_Diag\_And\_Groups\_inALLClaims.xlsx
        2. Unique\_Proc\_And\_Groups\_inALLClaims.xlsx
        3. Unique\_Drug\_And\_Groups\_inALLClaims.xlsx
     3. Output example:



1. **Get data (claims) source of patient IDs**
   * **Code:** 1\_HPC\_GetStudyIDSource.R
   * **Code description:** Record a patient has Medicare or Medicaid or both
   * **Input files:** 
     1. uh3\_kcrdata.csv
     2. kcr\_medicare\_claims\_fb0015.csv
     3. kcr\_medicaid\_healthclaims\_fb0015.csv
     4. KCR\_MEDICAID\_PHARMCLAIMS\_FB0015.csv
   * **Output:**
     1. Output dir: /1\_ID\_Sources\_Info/
     2. Output files: All\_ID\_Source.xlsx
     3. Output example:

|  |  |  |
| --- | --- | --- |
| Kcr\_ID | in\_Medicare | in\_Medicaid |
| 1 | 1 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 0 |

1. **Extract raw claims codes for each patient**
   * **Code:** 
     1. 2\_HPC\_GetPerPatientData\_Medicaid\_HealthClaims.R
     2. 2\_HPC\_GetPerPatientData\_Medicaid\_PharmClaims.R
     3. 2\_HPC\_GetPerPatientData\_Medicare.R
   * **Code description:** 
     1. Extract raw claims codes without cleaning for each patient in Medicaid or Medicare
     2. Raw claims codes are stored separately for individual patient and for different sources
   * **Input files:** 
     1. kcr\_medicaid\_healthclaims\_fb0015.csv
     2. KCR\_MEDICAID\_PHARMCLAIMS\_FB0015.csv
     3. kcr\_medicare\_claims\_fb0015.csv
   * **Output:**
     1. Output dir: / 2\_RawClaims\_perPatient/
     2. Output files: Stored for each patient
        1. PATIENTID\_all\_CLAIMSSOURCE\_claims.xlsx
     3. Output example:



1. **Get per-patient per-month data**
   * **Code:** 3A\_HPC\_Get\_PerMonthData\_withCleanCodes.R
   * **Code description:** 
     1. For each patient, clean codes from three sources (Medicaid health, Medicaid pharmacy, Medicare) claims data separately if available
     2. For each patient, compute month sequences using start and end date of claims data
     3. For each patient, extract unique codes in each month for all months.
     4. Construct patient per-Month data. Rows are months, Columns are all unique codes the patient had in all months. Entry = 1 represents the patient had the corresponding code in the corresponding month, Entry = blank for patient did not have the code in the month.
   * **Input files:** 
     1. 1\_All\_ID\_Source.xlsx
     2. 2\_RawClaims\_perPatient/Medicaid\_HealthClaims/
     3. 2\_RawClaims\_perPatient/Medicaid\_PharmClaims/
     4. 2\_RawClaims\_perPatient/Medicare/
   * **Output:**
     1. Output dir: /3\_CleanClaims\_perPatient\_perMonth/
     2. Output files: Stored for each patient
        1. PATIENTID\_perMonth\_Data.xlsx
     3. Output example:



1. **Get all cancer sites, dates, and types (primary or other or recurrence).**
   * **Code:** 4A\_Get\_Cancer\_SiteDateType.R
   * **Code description:** 
     1. For each patient, get all primary sites, other sites and recurrence dates
     2. Exclude patients does not have first primary information
     3. NOTE: if two events has the same dates, they merged together.
   * **Input files:** 
     1. uh3\_kcrdata.csv
   * **Output:**
     1. Output dir: / 4\_Recurr\_Dates\_Info/
     2. Output files: 4\_All\_cancer\_site\_date\_df.xlsx
     3. Output example:

|  |  |  |  |
| --- | --- | --- | --- |
| Site | Date | Type | study\_id |
| C504 | 12/03/2001 | First\_Primary | 277 |
| C446 | 02/06/2017 | Other | 277 |
| C508 | 05/17/2010 | First\_Primary | 570 |
| C508 | 01/02/2013 | Second\_Primary | 570 |
| C505 | 02/22/2005 | First\_Primary | 673 |
| C505 | 05/01/2008 | 1Recur | 673 |

1. **Get all event types, sites, and dates.**
   * **Code:** 4B\_GetEventType.R
   * **Code description:** 
     1. Extract three events types, site and dates as well as death date (Using date of last contact if cause of death info is not NA)
     2. Three events are defined as following:
        1. First event: 1st primary
        2. Second event:
           1. Recurrence of the 1st primary
           2. 2nd primary
           3. 1st primary breast cancer related death
        3. Subsequent cancer after the second event
     3. Exclude patients who has 2nd event within 6 months after 1st event.
   * **Input files:** 
     1. uh3\_kcrdata.csv
     2. uh3\_kcrdata\_add\_datelc.csv
     3. /4\_Recurr\_Dates\_Info/4\_All\_cancer\_site\_date\_df.xlsx
   * **Output:**
     1. Output dir: /4\_RecurrDates\_Outcome\_Info/
     2. Output files: 4\_All\_event\_df.xlsx
     3. Output example:



1. **Get patient-level outcome labels (has SBCE or not SBCE)**
   * **Code:** 4C\_Get\_BinaryOutcome.R
   * **Code description:** 
     1. A patient has any of the SBCEs is coded as 1, otherwise coded as 0.
     2. Second Breast Cancer Events (SBCE) is defined as follows:
        1. Recurrence of the 1st primary breast cancer
        2. The diagnosis of a 2nd primary breast cancer
        3. First primary BC related death
   * **Input files:** 4\_All\_event\_df.xlsx
   * **Output:**
     1. Output dir: /4\_RecurrDates\_Outcome\_Info/
     2. Output files: 4\_SBCE\_Label.xlsx
     3. Output example:

|  |  |
| --- | --- |
| study\_id | SBCE |
| 42 | 0 |
| 80 | 0 |
| 504 | 1 |

1. **Extract enrollment months for each patient.**
   * **Code:** 5A\_GetEnrollmentMonths.R
   * **Code description:** 
     1. For each patient, get a list of months that the patient has enrollment flag = 1
   * **Input files:** 
     1. kcr\_medicaid\_enroll\_fb0015.csv
     2. kcr\_medicare\_enroll\_fb0015.csv
   * **Output:**
     1. Output dir: /5\_Enrollment\_And\_Prediction\_Months/
     2. Output files: 5\_enrollment\_Months.xlsx
     3. Output example:

|  |  |
| --- | --- |
| study\_id | Enrolled\_Month |
| 1 | 1/1/00 |
| 1 | 2/1/00 |
| 1 | 3/1/00 |
| 1 | 4/1/00 |

1. **Get prediction start and end dates.**
   * Code: 5B\_GetPredictionMonth.R
   * Code description:
     1. For each patient, get pre-defined prediction start and prediction end dates
     2. Prediction starts and end dates are defined as follows:
        1. Start: the diagnosis date of the first primary breast cancer + 6 months
        2. End:
           1. Without SBCE: End of the enrollment
           2. With SBCE:

No third event: End of the enrollment

Has third event:

The third event is a non-breast primary cancer: 3 months before the diagnosis date

The third event is breast cancer event (Recurrence or diagnose of breast cancer or 1st primary BC death): 1 month before the diagnosis date

* + - 1. Compute prediction duration: prediction end - prediction start
    1. Exclude patients
       1. Enrollment end before prediction start
       2. Enrollment start after prediction end
       3. Prediction duration < 0 (e.g: 3rd event – 3 mon < 1st bc date + 6 mon)
       4. Prediction duration < 180 days (make sure patients has at least 6 month enrollment for prediction )
  + Input files:
    1. 4\_All\_event\_df.xlsx
    2. 5\_enrollment\_Months.xlsx
  + Output:
    1. Output dir: 5\_Enrollment\_And\_Prediction\_Months/
    2. Output files: 5\_prediction\_month\_df.xlsx
    3. Output example:



1. **Get per-Month data of enrolled months in prediction window**
   * Code: 6A\_HPC\_Get\_PerMonthData\_InTimeWindow\_AllEnrolls.R
   * Code description:
     1. For each patient, get enrolled per-Month data in prediction window
     2. New per-Month data:

Rows are months enrolled in prediction window. Columns are all unique codes the patient had in prediction window. Entry = 1 represents the patient had the corresponding code in the corresponding month, Entry = blank for patient did not have the code in the month.

* + 1. NOTE: for a particular enrollment month, patient might not have any of the codes. (e.g, the patient enrolled in this month, but did not have any codes in claims data)
  + Input files:
    1. PATIENTID\_perMonth\_Data.xlsx
    2. 5\_enrollment\_Months.xlsx
    3. 5\_prediction\_month\_df.xlsx
  + Output:
    1. Output dir: 6\_CleanClaims\_InValidMonth/EnrolledMonths\_WithPossibleMonthsHasNoCodes3/
    2. Output files: Stored for each patient
       1. PATIENTID\_perMonthData\_Enrolled\_inPredictionWindow.xlsx
    3. Output example:



1. **Get month-level outcome labels (Pre or Post SBCE)**
   * Code: 7A\_HPC\_Get\_PreOrPostLabels\_AllEnrolls.R
   * Code description:
     1. For each patient, each prediction month:
        1. the label is 0 if the month start date is before second event date
        2. the label is 1 if the month start date is after or equal second event date
     2. **NOTE:** If is possible the patient has SBCE, but no months has labeled 1 (e.g, no the post SBCE months or the SBCE month are not enrolled)
   * Input files:
     1. 4\_All\_event\_df.xlsx
     2. 4\_SBCE\_Label.xlsx
     3. PATIENTID\_perMonthData\_Enrolled\_inPredictionWindow.xlsx
   * Output:
     1. Output dir: /7\_PrePostLabels\_AndAvailibility6mon/A\_PrePost\_Labels/EnrolledMonths\_WithPossibleMonthsHasNoCodes/
     2. Output files: Stored for each patient
        1. PATIENTID\_PreOrPost\_MonthlyLabel.xlsx
     3. Output example:



1. **Check if patient has enough prediction months** 
   * Code: 7B\_HPC\_Get\_NumMonths\_And\_6MonAvalibility.R
   * Code description:
     1. For each patient, compute:
        1. Total number of prediction month
        2. Number of prediction months before second event
        3. Number of prediction months after or equal second event
        4. Flag indicating the patient has enough months in prediction window: “1” refers the patient has enough months, “0” refers the patient does not have enough months
     2. Enough number of prediction months definition:
        1. If no SBCE, at least 6 months
        2. If SBCE,
           1. If recurrence or diagnosis of 2nd primary, at least 3 months before and after 2nd event
           2. If death, at least have 3 months before death date
   * Input files:
     1. 4\_All\_event\_df.xlsx
     2. 4\_SBCE\_Label.xlsx
     3. PATIENTID\_PreOrPost\_MonthlyLabel.xlsx
   * Output:
     1. Output dir: 7\_PrePostLabels\_AndAvailibility6mon/
     2. Output files: NUM\_Months\_AvalFlags\_WithPossibleMonthsHasNoCodes.xlsx
     3. Output example:



1. **Get patient-level characteristics**
   * Code: 8A\_Get\_PatientLevelCharateristics.R
   * Code description:
     1. For each patient , get characteristics variables:
        1. List of variables: “Medicaid\_OR\_Medicare”, “SBCE”, “Diagnosis\_Year”, “Site”, “Race”, “BestStageGrp”, “Stage”, “Comb\_SEERSummStg”, “regional”, “Laterality”, "Grade", "er\_stat", "pr\_stat", "surg\_prim\_site\_V1", "surg\_prim\_site\_V2", "her2\_stat","radiation", "reg\_age\_at\_dx","reg\_nodes\_exam","reg\_nodes\_pos","cs\_tum\_size", "cs\_tum\_ext", "chemo", "hormone", "cs\_tum\_nodes","num\_nonbc", "date\_Birth", "DAJCC\_T","DAJCC\_M","DAJCC\_N", "Site\_1st\_Event","Date\_1st\_Event", "Site\_2nd\_Event","Type\_2nd\_Event","Date\_2nd\_Event", "Event\_2nd\_Is1stPrimaryBCDeath","Year\_1stPrimaryBCDeath","Days\_1stEventTODeath","Days\_1stTO2nd","Num\_Enrolled\_Prediction\_Months", "most\_recent\_enrollment\_year", "Num\_Month\_before\_2ndEvent", "Num\_Month\_AfterOrEqual\_2ndEvent","HasEnoughMonths\_InWindow"
        2. Note:
           1. Recode Race:

If race = 1, coded as 1

If race = 2, coded as 2

If race !=1 and race !=2, coded as 3

* + - * 1. Recode “BestStageGrp” stored in “Stage”:

Rules:

Stage 0: if BestStageGrp (0-2)

Stage 1: if BestStageGrp [10-30)

Stage 2: if BestStageGrp [30-50)

Stage 3: if BestStageGrp [50-70)

Stage 4 : if BestStageGrp 70-80)

* + - * 1. Recode SEERSummStg2000 stored in Comb\_SEERSummStg:

Rules:

Use “SEERSummStg2000” variable if it is not missing

If “SEERSummStg2000” is missing, use DerivedSS2000

* + - * 1. Compute DAJCC\_T/M/N variable using TNMPath and TNMClin:

Rules:

Use pathology results ('TNMPathT/M/N') if available (if not “NA” or “88” or “pX”), otherwise use clinical diagnosis results ('TNMClinT/M/N').

Convert DAJCC variable to numeric code, according to "pedsf\_attachment\_a.pdf",

* + - * 1. Code regional:

Rules:

If Comb\_SEERSummStg = 2,3,4,5, regional = 1

If Comb\_SEERSummStg != 2,3,4,5: regional = 0

If Comb\_SEERSummStg = NA, regional = NA

* + - * 1. Recode/group RXSummSurgPrimSite in two version, stored in surg\_prim\_site\_V1 and surg\_prim\_site\_V2:

Rules:

surg\_prim\_site\_V1: 0, 19, (20-24), 30, (40-42), (50-59,63), (60-62, 64-69, 73,74), 70-72, 80, 90, 99

surg\_prim\_site\_V2: 00,19,20 (21-24),30,40,41,42,50,51(53-56),52(57,58,59,63),60,61(64-67),62(68,69,73,74),70,71,72,80,90,99

43,44,45,46,47,48,49, 75 and 76 coded as NA

* + Input files:
    1. 4\_All\_event\_df.xlsx
    2. 4\_SBCE\_Label.xlsx
    3. uh3\_kcrdata.csv
    4. ky0015\_update\_DerivedSS2000\_andTNM.sas7bdat
    5. UH3 Nov Update with TNM and Staging.csv
    6. 1\_All\_ID\_Source.xlsx
    7. 4\_All\_cancer\_site\_date\_df.xlsx
  + Output:
    1. Output dir: 8\_Characteristics2/Patient\_Level/
    2. Output files: 8\_PatientLevel\_char\_WithPossibleMonthsHasNoCodes.xlsx
    3. Output example:



1. **Get Month-Level characteristics** 
   * Code: 8B\_Get\_MonthLevelChar\_AllEnrolls.R
   * Code description:
     1. For each patient each month, get characteristics
     2. List of variables: Enrolled\_year","Age","months\_since\_dx", "months\_to\_second\_event", "Race", "Site", "Stage","Grade","Laterality", "er\_stat","pr\_stat","her2\_stat", "surg\_prim\_site\_V1","surg\_prim\_site\_V2", "DAJCC\_T","DAJCC\_M","DAJCC\_N", "reg\_age\_at\_dx","reg\_nodes\_exam","reg\_nodes\_pos", "cs\_tum\_size","cs\_tum\_ext","cs\_tum\_nodes","regional"
     3. **NOTE:** Most of the variables are the same as patient-level, except the following ones: Enrolled\_year, Age, months\_since\_dx, months\_to\_second\_event
   * Input files:
     1. 8\_PatientLevel\_char\_WithPossibleMonthsHasNoCodes.xlsx
     2. PATIENTID\_PreOrPost\_MonthlyLabel.xlsx
   * Output:
     1. Output dir: 8\_Characteristics2/Month\_Level/MonthChar\_WithPossibleMonthsHasNoCodes/
     2. Output files: Stored for each patient
        1. PATIENTID\_MonthChar.xlsx
     3. Output example:



1. **Exclusion/inclusion of patients** 
   * Code: 9\_Get\_FinalIDs\_ByExclusions.R
   * Code description:
     1. Inclusion criterion:
        1. Stage 1,2,3
        2. 1st primary BC diagnosis year 2004 – 2015
        3. Local or regional stage (Comb\_SEERSummStg == 1 (Localized),2,3,4,5 (Regional))
        4. have enough enrolled months in prediction window, HasEnoughMonths\_InWindow == 1 (Refer to 7B\_HPC\_Get\_NumMonths\_And\_6MonAvalibility.R)
   * Input files:
     1. 8\_PatientLevel\_char\_WithPossibleMonthsHasNoCodes.xlsx
   * Output:
     1. Output dir: 9\_FinalIDs\_And\_UpdatedPtsChar/
     2. Output files:
        1. ID with char: 9\_PtsCharForFinalID\_WithPossibleMonthsHasNoCodes.xlsx (18239 Patients)
        2. IDs: 9\_Final\_ID1\_WithPossibleMonthsHasNoCodes.xlsx (18239 Patients)
     3. Output example:
        1. 9\_PtsCharForFinalID\_WithPossibleMonthsHasNoCodes.xlsx



* + - 1. 9\_Final\_ID1\_WithPossibleMonthsHasNoCodes.xlsx



1. **Get unique codes for each patient in prediction months**
   * Code: 10A\_Get\_PerPatient\_UniqueCodes\_AllEnrolls.R
   * Code description:
     1. For each patient, get the list of unique codes in prediction monnths
   * Input files:
     1. 9\_Final\_ID1\_WithPossibleMonthsHasNoCodes.xlsx
     2. PATIENTID\_perMonthData\_Enrolled\_inPredictionWindow.xlsx
   * Output:
     1. Output dir: 10A\_PerPatient\_UniqueCodes/WithPossibleMonthsHasNoCodes/
     2. Output files: Stored for each patient
        1. PATIENTID\_UniqueCodes.xlsx
     3. Output example:



1. **Get grouped diagnosis/procedure/drug codes features** 
   * Code:
     1. 10B\_Get\_PerMonth\_CCSDiag\_AllEnrolls.R
     2. 10C\_Get\_PerMonth\_CCSProc\_AllEnrolls.R
     3. 10D\_Get\_PerMonth\_DM3SPE\_AllEnrolls.R
     4. 10E\_Get\_PerMonth\_DM3GEN\_AllEnrolls.R
     5. 10F\_Get\_PerMonth\_ShortGNN\_AllEnrolls.R
   * Code description:
     1. For each patient, get CCS diagnosis, CCS procedure, DM3 specific drug, DM3 general drug, and short GNN drug grouped features
     2. For each patient, each month, the grouped feature refers to the sum of count of unique codes in the corresponding code group
     3. For each type of codes/groups feature (CCS diagnosis, CCS procedure, DM3 SEP drug, DM3 GEN drug and short GNN drug), they are stored in files separately for each patient. Rows refers to patient month, and columns (starting from the 5th) are unique code groups that patient has in the prediction window. Each entry refers to the sum of count of unique codes in the corresponding code group.
     4. NOTE: if patient does not have any group features, the corresponding file will only contain ID and month information
   * Input files:
     1. Unique\_Diag\_And\_Groups\_inALLClaims.xlsx
     2. Unique\_Proc\_And\_Groups\_inALLClaims.xlsx
     3. Unique\_Drug\_And\_Groups\_inALLClaims.xlsx
     4. PATIENTID\_perMonthData\_Enrolled\_inPredictionWindow.xlsx
   * Output:

Stored for each patient and each type of groups separately

* + 1. Output dir:
       1. 10B\_CCSDiagFeature\_inValidMonth/WithPossibleMonthsHasNoCodes/
       2. 10C\_CCSProcFeature\_inValidMonth/WithPossibleMonthsHasNoCodes/
       3. 10D\_DM3SPEFeature\_inValidMonth/WithPossibleMonthsHasNoCodes/
       4. 10E\_DM3GENFeature\_inValidMonth/WithPossibleMonthsHasNoCodes/
       5. 10F\_ShortGNNFeature\_inValidMonth/WithPossibleMonthsHasNoCodes/
    2. Output files:
       1. PATIENTID\_Month\_CCS\_DIAG\_Feature.xlsx
       2. PATIENTID \_Month\_CCS\_PROC\_Feature.xlsx
       3. PATIENTID \_Month\_DM3\_SPE\_Feature.xlsx
       4. PATIENTID \_Month\_DM3\_GEN\_Feature.xlsx
       5. PATIENTID \_Month\_S\_GNN\_Feature.xlsx
    3. Output example:

Only shows CCS Diagnosis group feature as example:



1. **Get frequency of code groups for SBCE and non-SBCE patient**
   * Code: 10G\_Count\_UniqueGrps\_forFinalIDsInValidMonths\_AllEnrolls.R
   * Code description:
     1. Compute number of patients has the at least one code in the group and the fraction of patients for SBCE and non-SBCE patient
   * Input files:
     1. 9\_PtsCharForFinalID\_WithPossibleMonthsHasNoCodes.xlsx
     2. Unique\_Diag\_And\_Groups\_inALLClaims.xlsx
     3. Unique\_Proc\_And\_Groups\_inALLClaims.xlsx
     4. Unique\_Drug\_And\_Groups\_inALLClaims.xlsx
     5. PATIENTID\_Month\_CCS\_DIAG\_Feature.xlsx
     6. PATIENTID \_Month\_CCS\_PROC\_Feature.xlsx
     7. PATIENTID \_Month\_DM3\_SPE\_Feature.xlsx
     8. PATIENTID \_Month\_DM3\_GEN\_Feature.xlsx
     9. PATIENTID \_Month\_S\_GNN\_Feature.xlsx
   * Output:
     1. Output dir: 10G\_Counts\_UniqueGrp\_PtsLevel/WithPossibleMonthsHasNoCodes/
     2. Output files:
        1. Count\_CCS\_Diag\_Unique\_Grps.xlsx
        2. Count\_CCS\_proc\_Unique\_Grps.xlsx
        3. Count\_DM3\_SPE\_Unique\_Grps.xlsx
        4. Count\_DM3\_GEN\_Unique\_Grps.xlsx
        5. Count\_S\_GNN\_Unique\_Grps.xlsx
     3. Output example:
        1. Only shows Count\_CCS\_Diag\_Unique\_Grps.xlsx as example:



1. **Select grouped code features as final model features by frequency**
   * Code: 10H\_SelectGroups.R
   * Code description:
     1. Select the groups that match or exceed the threshold of the fraction of patients with at least one code in that group, threshold are defined separately for SBCE and non-SBCE patient
     2. for non-SBCE, the thresholds are 0.15. and for SBCE patient thresholds are 0.10 for the diagnostic, procedure, and drug groups
   * Input files:
     1. Count\_CCS\_Diag\_Unique\_Grps.xlsx
     2. Count\_CCS\_proc\_Unique\_Grps.xlsx
     3. Count\_DM3\_SPE\_Unique\_Grps.xlsx
   * Output:
     1. Output dir: 10H\_Selected\_Grps/WithPossibleMonthsHasNoCodes/
     2. Output files:
        1. Selected\_CCSDiag\_Unique\_Grps.xlsx (130 groups)
        2. Selected\_CCSProc\_Unique\_Grps.xlsx (61 groups)
        3. Selected\_DM3SPEDrug\_Unique\_Grps.xlsx (27 groups)
     3. Output example:

Only shows selected ccs diagnosis unique groups as example:



1. **Get model-ready selected groups features (NUM: 218)**
   * Code: 11A\_Get\_ModelReady\_SelectedGroupFeature.R
   * Code description:
     1. For each patient, construct model ready data contains selected groups features.
     2. Rows are patient months, columns (Starting the 3rd) are all the unique selected groups features (218)
        1. 130 CCS diagnosis groups
        2. 61 CCS procedure groups
        3. 27 DM3 specific drug groups
   * Input files:
     1. 9\_Final\_ID1\_WithPossibleMonthsHasNoCodes.xlsx
     2. Selected\_CCSDiag\_Unique\_Grps.xlsx
     3. Selected\_CCSProc\_Unique\_Grps.xlsx
     4. Selected\_DM3SPEDrug\_Unique\_Grps.xlsx
     5. PATIENTID\_Month\_CCS\_DIAG\_Feature.xlsx
     6. PATIENTID \_Month\_CCS\_PROC\_Feature.xlsx
     7. PATIENTID \_Month\_DM3\_SPE\_Feature.xlsx
   * Output:
     1. Output dir: 11A\_ModelReady\_GrpFatures/WithPossibleMonthsHasNoCodes/
     2. Output files: Stored for patient separately
        1. PATIENTID\_Selected\_Grp\_Features.xlsx
     3. Output example:



1. **Get model-ready characteristic features (NUM: 23/58)**
   * Code: 11B\_Get\_ModelReady\_BinaryCharFeatures.R
   * Code description:
     1. Combine all patients’ month-level characteristic features into one with selected characteristic features
     2. Selected characteristic features (23) are:

"Enrolled\_year","Age","months\_since\_dx","Race" , "Site" , "Stage","Grade", "Laterality" ,"er\_stat","pr\_stat", "her2\_stat",

**"surg\_prim\_site\_V1", "surg\_prim\_site\_V2",** "DAJCC\_T" ,"DAJCC\_M","DAJCC\_N","reg\_age\_at\_dx","reg\_nodes\_exam", "reg\_nodes\_pos", "cs\_tum\_size", "cs\_tum\_ext",

"cs\_tum\_nodes", "regional" (NOTE: "surg\_prim\_site\_V1" and "surg\_prim\_site\_V2” are two version of surg\_prim\_site,, so two charisticaries feature data are generated separately for this two version, each includes one one version of surg\_prim\_site)

* + 1. Convert categorical features to binary feature columns: categorical features are "Race","Site","Stage","Grade","Laterality","er\_stat","pr\_stat","her2\_stat", "surg\_prim\_site\_V1","surg\_prim\_site\_V2"
    2. The output data also contains ID, month and month-level outcome information
    3. **NOTE:** in output data
       1. The number of columns is 61
       2. The number of info columns is 3
       3. After conversion to binary features, the number of features is 61 - 3 = 58
  + Input files:
    1. PATIENTID\_MonthChar.xlsx
    2. 9\_Final\_ID1\_WithPossibleMonthsHasNoCodes.xlsx
  + Output:
    1. Output dir: 11B\_ModelReady\_CharFatures/WithPossibleMonthsHasNoCodes/
    2. Output files:
       1. All\_Binary\_Chars\_WithSurgPrimSite\_V1.csv (NOTE: V1 results in better performance)
       2. All\_Binary\_Chars\_WithSurgPrimSite\_V2.csv
    3. Output example:



1. **Get model-ready transformation features (NUM: 654)**
   * Code: 11C\_Get\_ModelReady\_TransformationFeature.R
   * Code description:
     1. For each patient, each selected code group feature (218 selected features), compute three transformation features (654 transformation features)
     2. Transformation features are constructed as : Rows are patient-months, columns (starting from the 3rd ) are transformation features
     3. Transformation features are defined as follows:
        1. **time\_since :** 
           1. the time since the most recent occurrence of this code group
           2. Value set to -1 if patient has never got the code group till current time
        2. **time\_until**:
           1. the time since the soonest future occurrence of this code group
           2. Value set to -1 if patient is never going to get the code group after the current time
        3. **cumul\_ratio**:
           1. the total number of occurrences in each patient up to the time in question of that grouping divided by total elapsed time
           2. Value set to 0 or if patient has never got the code group till the current time. Value set to -1 for patient never got the code for all time.
   * Input files:
     1. 9\_Final\_ID1\_WithPossibleMonthsHasNoCodes.xlsx
     2. PATIENTID\_Selected\_Grp\_Features.xlsx
   * Output:
     1. Output dir: 11C\_ModelReady\_TransformFatures/WithPossibleMonthsHasNoCodes/
     2. Output files: Stored separately for each patient

PATIENTID\_Transf\_Features.xlsx

* + 1. Output example:



1. **Combine code groups, characteristics and transformation features (NUM: 930)**
   * Code: 11D\_Get\_ModelReady\_Combed\_Features.R
   * Code description:
     1. For each patient, combine selected code groups features, selected characteristic features, and transformation features
     2. The output data also contains sample id (patient id with month) and month-level outcome label (y\_PRE\_OR\_POST\_2ndEvent)
     3. The output data are formatted as: rows are patient-months, columns are all three types of features. (Except one for sample id and another one for label)
   * Input files:
     1. PATIENTID\_Selected\_Grp\_Features.xlsx
     2. All\_Binary\_Chars\_WithSurgPrimSite\_V1.csv
     3. PATIENTID\_Transf\_Features.xlsx
     4. 9\_Final\_ID1\_WithPossibleMonthsHasNoCodes.xlsx
   * Output:
     1. Output dir: 11D\_ModelReady\_CombFatures\_WithSurgPrimSite\_V1\_1201updated/ WithPossibleMonthsHasNoCodes/
     2. Output files: Stored for each patient seperatly

PATIENTID\_Comb\_Features.xlsx

* + 1. Output example:



1. **Combine all patients combined feature data into one**
   * Code: 11E\_Get\_AllPTs\_Comb\_Data.R
   * Code description:
     1. Combine all patients data into one data frame and saved it as “.rda” data.
   * Input files:
     1. PATIENTID\_Comb\_Features.xlsx
   * Output:
     1. Output dir: 11E\_AllPTs\_ModelReadyData/WithPossibleMonthsHasNoCodes/
     2. Output files: All\_PTS\_ModelReadyData.rda
     3. Output example: the same column format as PATIENTID\_Comb\_Features.xlsx, but rows contains all patients months data.
2. **Split data into train and test**
   * Code: 11F\_GetTrainTestIDs.R
   * Code description:
     1. To make sure no overlapping in original Ids in train and test:
        1. Test: 20% of the original IDs
        2. Train: 80% of the original IDs
        3. Validation sets will be auto generated when doing CV
   * Input files:
     1. 9\_Final\_ID1\_WithPossibleMonthsHasNoCodes.xlsx
     2. 9\_PtsCharForFinalID\_WithPossibleMonthsHasNoCodes.xlsx
   * Output:
     1. Output dir: 11F\_TrainTestIDs/
     2. Output files:
        1. test\_ID\_withLabel.xlsx
        2. train\_ID\_withLabel.xlsx
     3. Output example:

Only shows test ID for example:



1. **PCA analysis** 
   * Code: 12A\_PCA\_And\_tSNE.R
   * Code description:
     1. Run PCA analysis on Training data for non-categorical and non-constant features
     2. Get eigenvalues and variance explained for all PCA dimensions.
     3. Get variable contributions on each PCA dimension
     4. Plot 2D PCA
   * Input files:
     1. All\_PTS\_ModelReadyData.rda
     2. train\_ID\_withLabel.xlsx
     3. Unique\_Diag\_And\_Groups\_inALLClaims.xlsx (For adding group description)
     4. Unique\_Proc\_And\_Groups\_inALLClaims.xlsx (For adding group description)
   * Output:
     1. Output dir: 12A\_PCA\_TSNE\_Analysis/WithPossibleMonthsHasNoCodes/
     2. Output files:
        1. PCA\_Eigenvalues.csv
        2. PCA\_Explained\_Var.png
        3. PCA\_Variable\_Contribution.csv
        4. PCA\_2DPlot.png
     3. Output example:

PCA\_Variable\_Contribution.csv:



1. **Get top contributed features on PCA dim1 and dim2**
   * Code: 12B\_GetModel\_TrainData\_ForTopPCAFeatures.R
   * Code description:
     1. 1st approach: For training data, get top contributed features on most variance explained dimensions (dim1 and dim2).
        1. Top contributed features on dim 1 :
           1. cumul\_ratio\_CCS\_PROC\_202
           2. cumul\_ratio\_CCS\_PROC\_227
        2. Top contributed features on dim 2:
           1. months\_since\_dx
           2. Enrolled\_year (This one is exclude for later analysis)
        3. Output data saved in “rda”, rows are patient months, columns (starting from the 3rd column) are four top contributed features and month-level label
     2. 2nd approach: For training data, compute the weighted sum of normalized top 10 contributed features on dim 1
        1. Min-max normalize top 10 features on dim 1
        2. For each patient month, the weighted sum score is computed as:
        3. Output data are saved in both “rda” and “csv”, rows are patient month, columns (starting from the 3rd ) weighted sum scores and normalized top 10 features
   * Input files:
     1. All\_PTS\_ModelReadyData.rda
     2. train\_ID\_withLabel.xlsx
     3. PCA\_Variable\_Contribution.csv
   * Output:
     1. Output dir: 12B\_TopPCAFeature\_ModelReady\_TrainData/WithPossibleMonthsHasNoCodes/
     2. Output files:
        1. 4F\_ModelReady\_TrainData.rda
        2. Dim1\_Weighted\_Sum\_feature.csv/Dim1\_Weighted\_Sum\_feature.rda
     3. Output example:

****

1. **Distribution plot for training data**
   * Code: 12C\_DistributionPlotByLabel.R
   * Code description:
     1. For training data , plot distribution (boxplot, violin plot and histogram) for both “pre” and “post” samples
   * Input files: 4F\_ModelReady\_TrainData.rda
   * Output:
     1. Output dir: 12C\_Train\_DistributionPlot/WithPossibleMonthsHasNoCodes/
     2. Output files:
        1. Box\_plot/
        2. Histogram/
        3. Violin\_plot
     3. Output example:

Chart, box and whisker chart

Description automatically generated Chart, histogram

Description automatically generated Chart

Description automatically generated

1. **Determine selection criteria for obvious positive and negative samples from training data**
   * Code: 12D\_FindSamples\_BySelectionCriteria.R
   * Code description:
     1. From training data , determine selection criteria by trying different combinations of feature values thresholds for top three contributed feature to:
        1. minimize the negative to positive ratio in non-obvious samples
        2. maximining the precision of negative/positive samples in selected samples (named obvious negative/obvious positive samples)
           1. For obvious negative samples,

Best combination:

cumul\_ratio\_CCS\_PROC\_202 = -1

cumul\_ratio\_CCS\_PROC\_227 < 0.2

months\_since\_dx < 36

* + - * 1. For obvious positive samples,

Best combination:

0.2 < cumul\_ratio\_CCS\_PROC\_202 < 0.4

2 < cumul\_ratio\_CCS\_PROC\_227 < 4

60 < months\_since\_dx < 90

* + - 1. Plot NEG-to-POS ratio vs. precision of negative/ positive samples
      2. Identify obvious negative and positive samples in training data using the best combination of thresholds, as well as the non-obvious samples
  + Input files:
    1. 4F\_ModelReady\_TrainData.rda
  + Output:
    1. Output dir: 12D\_ExclusionSamples/WithPossibleMonthsHasNoCodes/Train/
    2. Output files:
       1. Precision\_Ratio\_Plot\_NEG.png
       2. Precision\_Ratio\_Plot\_POS.png
       3. ObviousPos\_Samples.csv
       4. ObviousNeg\_Samples.csv
       5. NON\_Obvious\_Samples.csv
    3. Output example:

Shows ObviousPos\_Samples.csv as example:



1. **Identify obvious negative, positive samples and non-obvious samples (testing data)**
   * Code: 12E\_FindSamples\_BySelectionCriteria\_Test.R
   * Code description:
     1. Identify obvious negative and positive samples in testing data using the best combination of thresholds determined by training data, as well as the non-obvious samples
   * Input files:
     1. test\_data.rda (This file is generated from 15\_XGBoost\_InputData.R)
     2. **Or** All\_PTS\_ModelReadyData.rda and test\_ID\_withLabel.xlsx)
   * Output:
     1. Output dir: 12D\_ExclusionSamples/WithPossibleMonthsHasNoCodes/Test/
     2. Output files:
        1. ObviousNeg\_Samples\_Test.csv
        2. ObviousPOS\_Samples\_Test.csv
        3. NON\_Obvious\_Samples\_Test.csv
     3. Output example:

Shows ObviousPOS\_Samples\_Test.csv as example:



1. **Generate training and testing data**
   * Code: 15\_XGBoost\_InputData.R
   * Code description:
     1. Generate XGBoost input data
        1. Test: all testing sample
        2. Train: non-obvious samples
           1. Option1: all non-obvious samples without sampling
           2. Option2: Down sample 10 times
   * Input files:
     1. All\_PTS\_ModelReadyData.rda
     2. Train/NON\_Obvious\_Samples.csv
     3. test\_ID\_withLabel.xlsx
     4. ConstFeature\_removed\_ForPCAandtSNE.csv
   * Output:
     1. Output dir: 15\_XGB\_Input/
     2. Output files:
        1. test\_data.rda
        2. train\_data\_DSNUM.rda, TrainSampleIDsAndLabels\_NUM.csv
           1. NUM = 0 : without down sampling
           2. NUM = 1-10: one of the down sample datasets
2. **Run XGBoost**
   * Code: 16\_Run\_Xgboost.R
   * Code description:
     1. Run XGboost 11 times for each training data (one without down sampling and 10 down sampled )
     2. For each run using each training dataset,
        1. First use BayesianOptimization and 10-fold cross-validation on training data to get optimal hyperparameters
        2. Use the optimal hyperparameter, then train the optimal model
        3. Record important features find by optimal model
        4. Get prediction probabilities using optimal model for all testing data and compute performance (cutoff=0.5)
   * Input files:
     1. test\_data.rda
     2. train\_data\_DSNUM.rda
   * Output:
     1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/All\_DS\_Performance/
     2. Output files:

Stored for each train dataset: NUM refers to each train dataset

* + - 1. train\_DSNUM/ BeforeSmoothed/
         1. 16\_importance\_matrix\_DSNUM.csv
         2. 16\_Prediction\_Table\_DSNUM.csv
         3. 16\_Performance\_Table\_DSNUM.csv
    1. Output example:
       1. Importance matrix:



* + - 1. Prediction Table:



* + - 1. Performance table:



1. **Re-Run XGBoost using importance features** 
   * Code: 16\_Run\_Xgboost2\_Topfeatures.R
   * Code description:
     1. The same as 16\_Run\_Xgboost.R except only use top 50 features.
   * Input files:
     1. test\_data.rda
     2. train\_data\_DSNUM.rda
     3. 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/All\_DS\_Performance/train\_DSNUM/BeforeSmoothed/16\_importance\_matrix\_DSNUM.csv
   * Output:
     1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance/
     2. Output files:
        1. train\_DSNUM/ BeforeSmoothed/
           1. 16\_OptimalModelParam.csv
           2. modelNUM.rds
           3. 16\_importance\_matrix\_DSNUM.csv
           4. 16\_Prediction\_Table\_DSNUM.csv
           5. 16\_Performance\_Table\_DSNUM.csv
     3. Output example: Similar to 34. 16\_Run\_Xgboost.R
2. **Prediction of obvious negative and positive training samples**
   * Code: 16B\_Prediction\_of\_ObvNegPosTrainData.R
   * Code description:
     1. For each obvious sample, predict using two methods:
        1. Predict using optimal model
        2. Predict all as negative/positive
   * Input files:
     1. Trained model 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance/train\_DSNUM//BeforeSmoothed/modelNUM.rds
     2. All patients model ready data

All\_PTS\_ModelReadyData.rda

* + 1. Training obvious samples IDs:
       1. 12D\_ExclusionSamples/WithPossibleMonthsHasNoCodes/Train/ ObviousNeg\_Samples.csv
       2. 12D\_ExclusionSamples/WithPossibleMonthsHasNoCodes/Train/ ObviousPos\_Samples.csv
  + Output:
    1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Prediction\_obvTrainData/
    2. Output files: Stored separately for each trained model
       1. train\_DSNUM/
          1. pred\_tb\_obvNEG.csv
          2. perf\_tb\_obvNEG.csv
          3. pred\_tb\_obvPOS.csv
          4. perf\_tb\_obvPOS.csv
    3. Output example:

Shows results for negative as example:

* + - 1. pred\_tb\_obvNEG.csv



* + - 1. perf\_tb\_obvNEG.csv



1. **Update predictions for obvious positive and negative testing samples**
   * Code: 16C\_UpdatePrediction\_TestData.R
   * Code description:
     1. For obvious negative samples, update the prediction as 0
     2. For obvious positive samples, update the prediction as 1
     3. For non-obvious samples, keep the prediction using the optimal model
   * Input files:
     1. test\_data.rda
     2. testing data sample IDs by group:
        1. ObviousNeg\_Samples\_Test.csv
        2. ObviousPos\_Samples\_Test.csv
        3. NON\_Obvious\_Samples\_Test.csv
     3. Prediction table using optimal model:

16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance/

train\_DSNUM/BeforeSmoothed/ 16\_Prediction\_Table\_DSNUM.csv

* + Output:
    1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance/
    2. Output files:

train\_DSNUM/BeforeSmoothed/ 16\_Updated\_Prediction\_Table\_DSNUM.csv

* + 1. Output example:

“pred” columns stores the updated predicted value

****

1. **Change point analyses** 
   * Code: 16D\_ChangePointAnalysis.R
   * Code description:
     1. Change point analysis for two prediction results. (one is from using optimal model to predict all test cases, another is from updated prediction for obvious cases)
     2. Identifying changes in mean and variance of predicted probabilities using binary segmentation using cpt.meanvar(). Functions adopted from plot\_changepoint\_info() from src/Tomas/run\_xgboost.s3.r. The maximum number of changepoint is set to 1.
   * Input files:
     1. 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance//train\_DSNUM/BeforeSmoothed/
        1. Optimal model:

16\_Prediction\_Table\_DSNUM.csv

* + - 1. Updated model:

16\_Updated\_Prediction\_Table\_DSNUM.csv

* + 1. Test ID and labels:

11F\_TrainTestIDs/test\_ID\_withLabel.xlsx

* + Output:
    1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance/
    2. Output files:

train\_DSNUM/ BeforeSmoothed/

* + - 1. Optimal model:

16\_ Original\_ChangePoint.csv

* + - 1. Updated model:

16\_ Updated\_ChangePoint.csv

* + 1. Output example:

Shows 16\_ Original\_ChangePoint.csv as example:



1. **Compute performance for optimal trained model and updated model**
   * Code: 16E\_Compute\_Perf.R
   * Code description:
     1. Compute month-level and patient-level performance
     2. For month-level performance:
        1. Compute AUC, Accuracy, Recall, Specificity, Precision, F1, NPV, FPR and FNR using 8 different cutoff thresholds 0.1,0.2,0. 3, …, 0.8 for subsets of testing data
        2. Subsets of testing data:
           1. All testing samples
           2. Obvious negatives samples
           3. Obvious positive samples
           4. Non-obvious samples
           5. Random sample testing data (balanced and imbalanced) 5 times for each ratio (POS: NEG = 1:1, POS: NEG = 1:2, POS: NEG = 1:5)

For each sampled testing data, compute AUC, Accuracy, Recall, Specificity, Precision, F1, NPV, FPR and FNR using 8 different cutoff thresholds 0.1,0.2,0. 3,.., 0.8.

For all sampled testing data in each ratio category, report the average performance

* + 1. For patient-level performance:
       1. For each patient, compute the predicted month of SBCE for each threshold
          1. Predicted month of SBCE is defined in two ways:

The first month that the prediction probability is greater or equal to the prediction probability threshold

The first month of 3 consecutive months that predicts probability greater or equal to the prediction probability threshold

The month generated from change point analysis. (Refer to 16D\_ChangePointAnalysis.R)

* + - * 1. Compute absolute difference between predicted month and actual SBCE month for each threshold (For SBCE patients)

compute mean, std, median, Q1, Q3,min and max

* + - 1. For each patient, get the predicted class (SBCE or non-SBCE):
         1. if there exists a predicted month, then the patient is predicted as SBCE patient, coded as 1. Otherwise, the patient predicted outcome is non-SBCE patient, coded as 0.
         2. Compute Accuracy, Recall, Specificity, Precision, F1, NPV, FPR and FNR using 8 different cutoff thresholds 0.1,0.2,0. 3, …, 0.8. (**NOTE**: since there is only predicted class, no probability, so there is no AUC)
  + Input files:
    1. Prediction table: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance//train\_DSNUM/BeforeSmoothed/
       1. Optimal model:

16\_Prediction\_Table\_DSNUM.csv

* + - 1. Updated model:

16\_Updated\_Prediction\_Table\_DSNUM.csv

* + 1. Change point results:

16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance/

train\_DSNUM/ BeforeSmoothed/

* + - 1. Optimal model:

16\_ Original\_ChangePoint.csv

* + - 1. Updated model:

16\_ Updated\_ChangePoint.csv

* + 1. Testing sample ID by groups:

12D\_ExclusionSamples/WithPossibleMonthsHasNoCodes/Test/

* + - 1. ObviousNeg\_Samples\_Test.csv
      2. ObviousPos\_Samples\_Test.csv
      3. NON\_Obvious\_Samples\_Test.csv
    1. Patient-level characteristics

8\_PatientLevel\_char\_WithPossibleMonthsHasNoCodes.xlsx

* + Output:
    1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance/train\_DSNUM/BeforeSmoothed/
    2. Output files: (Original/Updated refers to the optimal trained model or obvious-cases approach)
       1. Month-level performances:
          1. All testing samples:

Original/Updated\_perf\_tb\_alltest.csv

* + - * 1. Obvious negatives samples

Original/Updated \_perf\_tb\_alltestNEG.csv

* + - * 1. Obvious positive samples

Original/Updated \_perf\_tb\_alltestPOS.csv

* + - * 1. Non-obvious samples

Original/Updated \_perf\_tb\_alltestNONOBV.csv

* + - * 1. Random sample testing data

Original/Updated \_perf\_tb\_pos1\_neg1.csv

Original/Updated \_perf\_tb\_pos1\_neg2.csv

Original/Updated \_perf\_tb\_pos1\_neg5.csv

* + - 1. Patient-level:
         1. Predicted months:

Original/Updated\_Patient\_Level\_PerdMonth\_alltest.csv

Original/Updated\_ Patient\_Level\_PerdMonth\_alltest\_persistent3month.csv

Change point analysis results in previous output.

* + - * 1. Performance:

Original/Updated \_Patient\_Level\_Perf\_tb\_alltest.csv

Original/Updated\_ Patient\_Level\_Perf\_tb\_alltest\_persistent3month.csv

Change point analysis results is not used for performance

1. Month difference performance:
   1. Original/Updated\_ Predicted\_Monthdiff\_stats.csv
   2. Original/Updated\_ Predicted\_Monthdiff\_stats\_persistent3month.csv
2. **Plot prediction trajectory for each patient**
   * Code: 16F\_PlotPredictions.R
   * Code description:
   * Input files:
     1. Prediction table: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance//train\_DSNUM/BeforeSmoothed/
        1. Optimal model:

16\_Prediction\_Table\_DSNUM.csv

* + - 1. Updated model:

16\_Updated\_Prediction\_Table\_DSNUM.csv

* + 1. Test ID and labels: test\_ID\_withLabel.xlsx
    2. Predicted months:
       1. Original/Updated\_Patient\_Level\_PerdMonth\_alltest.csv
       2. Original/Updated\_ Patient\_Level\_PerdMonth\_alltest\_persistent3month.csv
       3. 16\_ Original/Updated\_ChangePoint.csv
  + Output:
    1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance/train\_DSNUM//BeforeSmoothed/Individual\_Plot/
    2. Output files: Stored for each patient separately

SBCE0/1\_PATIENTID.png

1. **Smooth out the predicted probability by curve fitting**
   * Code: 16G\_CurveFitting.R
   * Code description:
     1. For each patient, use logistic curve fitting to get smoothed predicted probability
   * Input files:
     1. Prediction table: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/Use\_ImportantFs\_Performance//train\_DSNUM/BeforeSmoothed/
        1. Optimal model:

16\_Prediction\_Table\_DSNUM.csv

* + - 1. Updated model:

16\_Updated\_Prediction\_Table\_DSNUM.csv

* + 1. Test ID and labels:
       1. test\_ID\_withLabel.xlsx
  + Output:
    1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1111updated/All\_DS\_Performance/AfterSmoothed/ (NOTE: this output is not updated for most recent prediction table due to lower performance previously)
    2. Output files: 16\_Smoothed\_Prediction\_Table\_DS1\_posweight0.5.csv
    3. Output example:

****

1. **Compute performance after curve fitting**
   * Code: 16H\_Compute\_AfterSmooth\_Perf.R
   * Code description:
     1. The procedures are as 16E\_Compute\_Perf.R, except using different input
     2. **NOTE**: This code is not updated for most recent prediction table, and it can be integrated into 16E\_Compute\_Perf.R
   * Input files:
     1. 16\_Performance\_WithSurgPrimSite\_V1\_1111updated/All\_DS\_Performance/AfterSmoothed/16\_Smoothed\_Prediction\_Table\_DS1\_posweight0.5.csv
   * Output:
     1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1111updated/All\_DS\_Performance/AfterSmoothed
     2. Output files: Similar to output in 16E\_Compute\_Perf.R
2. **Plot prediction trajectory after curve fitting**
   * Code: 16I\_PlotPrediction\_AfterSmooth.R
   * Code description:
     1. The procedures are as 16E\_Compute\_Perf.R, except using different input
     2. **NOTE**: This code is not updated for most recent prediction table, and it can be integrated into 16E\_Compute\_Perf.R
   * Input files:
     1. 16\_Performance\_WithSurgPrimSite\_V1\_1111updated/All\_DS\_Performance/AfterSmoothed/16\_Smoothed\_Prediction\_Table\_DS1\_posweight0.5.csv
     2. test\_ID\_withLabel.xlsx
   * Output:
     1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1111updated/All\_DS\_Performance/AfterSmoothed/Individual\_Plot/
3. **Compute descriptive statistics**
   * Code: 17\_ComputeDispStats.R
   * Code description:
     1. Report patient-level number of patients and SBCE status and month-level numbers of samples and pre/post status in :
        1. All Training
        2. All Testing
        3. obvious neg training
        4. obvious pos training
        5. non-obvious training
        6. obvious neg test
        7. obvious pos test
        8. non-obvious test
     2. Compute patient-level descriptive statistics for:
        1. All patients
        2. SBCE patients
        3. Non-SBCE patients
     3. Plot patient-level histograms
   * Input files:
     1. 9\_PtsCharForFinalID\_WithPossibleMonthsHasNoCodes.xlsx
     2. All\_PTS\_ModelReadyData.rda
     3. train\_ID\_withLabel.xlsx
     4. test\_ID\_withLabel.xlsx
     5. Train/ObviousNeg\_Samples.csv
     6. Train/ObviousPos\_Samples.csv
     7. Train/NON\_Obvious\_Samples.csv
     8. Test/ObviousNeg\_Samples\_test.csv
     9. Test/ObviousPos\_Samples\_test.csv
     10. Test/NON\_Obvious\_Samples\_test.csv
   * Output:
     1. Output dir: 17\_Discrip\_Statistics\_1217Updated/
     2. Output files:
        1. discrip\_table.csv
        2. Diagnosis Year and most recent enrollment histograms.
     3. Output example:



1. **Check patients’ feature trajectory who has sudden jump in predictions**
   * Code: 18\_CheckPTsFeatureTrajectory.R
   * Code description:
     1. For patient who has sudden jumps in prediction, find list of features which also has sudden increased or decreased values before predicted probability jump
   * Input files:
     1. PATIENTID\_Comb\_Features.xlsx
     2. 16\_Prediction\_Table\_DSNUM.csv
     3. Unique\_Diag\_And\_Groups\_inALLClaims.xlsx
     4. Unique\_Proc\_And\_Groups\_inALLClaims.xlsx
   * Output:
     1. Output dir: /ReCAPSE\_Intermediate\_Data/recapse\_checkPts
     2. Output files:

PATIENTIDfeature\_changed.csv

* + 1. Output example:



1. **SHAP analysis** 
   * Code: 19\_SHAP.R
   * Code description:
     1. Get SHAP summary plot using optimal model for
        1. Training data
        2. Testing data
        3. Obvious-negative testing data
        4. Obvious-positive testing data
        5. Non-obvious testing data
   * Input files:
     1. Optimal model:

train\_DSNUM/BeforeSmoothed/modelNUM.rds

* + 1. Optimal model important features:

train\_DSNUM/BeforeSmoothed/16\_importance\_matrix\_DSNJM.csv

* + 1. Train and test data:
       1. train\_data\_DSNUM.rda
       2. test\_data.rda
    2. Test IDs in different groups:
       1. Test/ObviousNeg\_Samples\_Test.csv
       2. Test/ObviousPOS\_Samples\_Test.csv
       3. Test/NON\_Obvious\_Samples\_Test.csv"
  + Output:
    1. Output dir: 16\_Performance\_WithSurgPrimSite\_V1\_1217updated/SHAP/

1. **Run simple baseline algorithms for prediction of test data**
   * Code: 21\_BaselineAlg.R
   * Code description:
     1. Implemented three baseline algorithms:
        1. Random guess: Random generate 0 or 1 to each sample
        2. All 1s: Predicted 1s for all samples
        3. All 0s: Predicted 0s for all samples
   * Input files:
     1. test\_data.rda
   * Output:
     1. Output dir: 21\_BaselineResults/
     2. Output files: perf\_baseline.csv
     3. Output example:



1. **Utility functions Code**
   * Code: Recapse\_Ultility.R
   * Code description:
     1. This script contains all functions used throughout all codes for RECAPSE Projects

**Updates:**

1. **Found duplicates drug codes, e.g, 00003011675 is the same as 3011675**
   1. Updated clean\_code\_func2() in Recapse\_Ultility.R to remove leading 0s
   2. Direct affected code and outputs:

|  |  |  |
| --- | --- | --- |
| Code | Output file | Status |
| 0B\_CleanUniqueCodes.R | 0\_Cleaned\_Unique\_Drug\_Codes.xlsx | Fixed |
| 0C\_Code\_Grouping.R | 1. Unique\_Drug\_And\_Groups\_inALLClaims.xlsx 2. Codes\_Stats\_inALLClaims\_updated.csv | Fixed |
| 3A\_HPC\_Get\_PerMonthData\_withCleanCodes.R |  |  |

* 1. **48942 -> 41271 unique drug codes**